ปีการศึกษา 2533

ระบบมิดี้
MIDI SYSTEM

โดย

ประสิทธิ์ วรฉัตราวณิช
อุทัย เกียรติวิกรัย

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
0228780
12 มิ.ย. 2534

ปริญญานิพนธ์ปีการศึกษา 2533

เรื่อง ระบบมิดี้

ผู้จัดทำ

1. นาย ประสิทธิ์ วรฉัตราวณิช 301148
2. นาย อุทัย เกียรติวิกรัย 301367

.....อาจารย์ที่ปรึกษา

(ผศ. พลผดุง ผดุงกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028780

ระบบมิดี้
(MIDI SYSTEM)

ประสิทธิ์ วรฉัตราวณิช 301148
อุทัย เกียรติวิกรัย 301367

อาจารย์ที่ปรึกษา
ผศ.พลผดุง ผดุงกุล
ปีการศึกษา 2533

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ เสนอการนำเครื่องคอมพิวเตอร์มาช่วยในงานด้าน ศิลปทางดนตรี ซึ่งเป็นงานที่มีความละเอียดอ่อนอย่างมาก โดยมีจุดประสงค์ให้เป็นประโยชน์กับผู้ที่ต้องการศึกษาทางด้านดนตรี หรือช่วยงานผู้ที่มีความชำนาญในงานนี้อยู่แล้ว โครงงานนี้จะทำให้ ไมโครคอมพิวเตอร์ (MICROCOMPUTER) ทำงานควบคุมเครื่องดนตรีต่าง ๆ ให้เล่นเพลงตามต้องการโดยการบันทึกการเล่นของเครื่องดนตรีแต่ละชิ้นเข้าไปก่อน แล้วจึงส่งข้อมูลทั้งหมดให้เครื่องดนตรีเล่นเป็นเพลงพร้อมกัน โดยในขณะที่มีการรับข้อมูลเข้าไป ไมโครคอมพิวเตอร์ สามารถส่งข้อมูลเก่าที่เก็บไว้แล้วออกไปพร้อมกันด้วย รวมทั้งการปรับปรุงแก้ไขข้อมูลดังกล่าวในขณะที่บันทึกด้วย นอกจากนั้น ไมโครคอมพิวเตอร์ ยังทำหน้าที่เป็น กลองอิเล็คทรอนิคส์ (DRUM MACHINE)ด้วยซึ่งผู้ใช้สามารถโปรแกรมผ่าน มอนิเตอร์ (MONITOR) ได้อย่างสะดวก การแก้ไขข้อมูลที่บันทึกเข้าไปก็สามารถกระทำได้โดยสะดวก ด้วยวิธีการแสดงผลเป็นรูปการกด คีย์ (KEY) หรือการเล่นในขณะนั้น ๆ ,การลบ,การ COPY และเลื่อนระดับเสียง ด้วยเช่นกัน นอกจากนี้ยังมีส่วน ควอนไตซ์ (QUANTIZE)ซึ่งจะทำหน้าที่วิเคราะห์ข้อมูลที่เล่นจากนักดนตรีว่าควรจะลงจังหวะอย่างไรถึงจะถูกต้อง ช่วยให้ประสิทธิภาพในการบันทึกสมบูรณ์ยิ่งขึ้น ในด้านการเชื่อมต่อได้สร้างคาร์ดที่สามารถรับและส่งข้อมูลที่เป็นมาตรฐานทางมิดี้ได้ รายละเอียดต่าง ๆ ของทั้งส่วนที่เป็น ฮาร์ดแวร์ และซอฟท์แวร์ ได้นำเสนออย่างละเอียดแล้วในวิทยานิพนธ์ฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDI SYSTEM

Prasit Worachuttrawanit
Uthai Kiattivikrai

Advisor
Mr. Polpadung Padungkul

In this thesis , the application of computer aided in art of music , the extreme delicate work , is proposed not only for the one who interests in this area , but the professional one also. This project is carried out on the feasibility of employing microcomputer to control the synthesis of various musical instruments as required. This technique is initially acheived by recording the MIDI data produced by each musical instruments. After that , the overall data which are stored in microcomputer will be transmitted to each instrument. Thus , the overall sound can be generated simultaneously. By computerized control , computer is able to send out data and/or update data while receiving data as well. Moreover , microcomputer is also served as a drum machine , the convenient tool for user to program data via monitor. Updating data can be made easily by displaying the figure of input key at that time. Deleting , copying and transposing sound are developed also. In addition , quantize is included to analyse input data , that is , to acquire the appropriate of musical arrangement and to make recording more efficiently. Concerning with interfacing data , interfaced card is implemented for communicating data , in accordance with the MIDI standard. The description of hardware and software is presented in details in this thesis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง		หน้า
	บทคัดย่อภาษาไทย	ก
	บทคัดย่อภาษาอังกฤษ	ข
	สารบัญ	ค
บทที่ 1	บทนำ	ง
บทที่ 2	ทฤษฎีของระบบมิดี้	1
บทที่ 3	การออกแบบและการสร้าง	12
บทที่ 4	การทดลองและผลการทดลอง	20
บทที่ 5	สรุปและวิจารณ์	23
	กิตติกรรมประกาศ	24
	ภาคผนวก	
	MIDI - Musical Instrument Digital Interface	25
	MIDI Implementation Chart	32
	หนังสืออ้างอิง	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

ปัจจุบัน คอมพิวเตอร์ได้เข้ามาเป็นส่วนหนึ่งของชีวิตประจำวันของเรา มากขึ้นเรื่อย ๆ ไม่ว่าในด้านการศึกษา ด้านธุรกิจ ด้านอุตสาหกรรม หรือ แม้กระทั่งในเรื่องของศิลปะทางด้านดนตรี

คอมพิวเตอร์ เริ่มเข้ามาเกี่ยวพันธ์กับศาสตร์ทางด้านดนตรี ในระยะที่มีการสร้างเครื่องดนตรีพวก ดรัมแมชชีน(DRUMMACHINE) ซึ่งทำหน้าที่เป็นผู้ตีกลอง โดยในยุคแรกยังคงเป็นเพียงวงจรนับ กับ วงจรตรรกธรรมดา ต่อมาในปี 1980 บริษัทผู้ผลิตซินธิไซเซอร์(SYNTHESIZER) ได้มีการนำ ไมโครโปรเซสเซอร์(MICROPROCESSOR) มาเป็นตัว อ่าน คีย์บอร์ด แล้วนำข้อมูลส่งให้ ออสซิลเลเตอร์(OSCILLATOR) สร้างเป็นความถี่ตาม โน๊ตดนตรีที่กด นับจากนั้นเป็นต้นมาได้มีการพัฒนาประสิทธิภาพของเครื่องดนตรีอย่างต่อเนื่อง บนพื้นฐานของระบบคอมพิวเตอร์ จนกระทั่งในปี 1983 ได้มีการนำข้อมูลการอ่านคีย์บอร์ด ส่งออกจากตัวเครื่องดนตรีด้วย โดยบริษัทผู้ผลิตเครื่องดนตรีได้รวมตัวกันจัดตั้งมาตรฐานในการส่งข้อมูลนี้ ในชื่อว่า มิดี้ (MIDI 1.0MUSICAL INSTRUMENT DIGITAL INTERFACING)

นับจากวันนั้นจนถึงวันนี้ ระบบ มิดี้ ได้สร้างประโยชน์ต่องานทางด้านดนตรีอย่างสูง ระบบ มิดี้ ทำให้มีการสื่อสารกันระหว่าง เครื่องดนตรีกับเครื่องดนตรี และเครื่องดนตรีกับ ซีเควนเซอร์(SCQUENCER) ซึ่งเป็นอุปกรณ์บันทึกข้อมูลมิดี้ ไปจนถึงการสื่อสารระหว่างเครื่องดนตรีกับ ไมโครคอมพิวเตอร์ สำหรับโครงงานนี้เป็นการนำเอา ไมโครคอมพิวเตอร์ตระกูล IBM PC มาใช้ติดต่อสื่อสารข้อมูลกับเครื่องดนตรี

<u>ประโยชน์ที่ได้จากโครงงานนี้</u>

- จะทำให้เราสามารถควบคุม เครื่องดนตรี ให้เล่นเพลงให้กับเรา ตามที่เราต้องการได้ ทำให้นักดนตรีสามารถแต่งเพลง และสร้างงานทางด้านดนตรีได้ รวดเร็วและมีประสิทธิภาพ กว่าการใช้ความสามารถที่มีอยู่ในตัวนักดนตรีเอง และนอกจากนั้นยังมีประโยชน์ต่อผู้ฝึกหัดเล่นดนตรีใหม่รวมไปถึงผู้ที่สนใจ เพราะเราสามารถนำข้อมูลเพลงจากนักดนตรีที่มีความสามารถ มาศึกษาได้
- ทำให้การ แต่งเพลง สามารถทำได้โดยง่ายกว่าเก่ามาก ซึ่งผู้แต่งก็ไม่จำเป็นที่ต้องมีความสามารถทางด้านดนตรีสูงและไม่จำเป็นที่จะต้องเล่นดนตรีเป็นหลาย ๆ ประเภทในการแต่งเพลง หนึ่งๆ
- ทำให้ประหยัด เครื่องดนตรี และอุปกรณ์ ลงไปมาก เพราะ ไมโครคอมพิวเตอร์ (MICROCOMPUTER) ตัวเดียว สามารถทำหน้าที่แทนอุปกรณ์ได้หลายชนิด
- เป็นพื้นฐานในการศึกษาถึง ระบบมิดี้(MIDI) เนื่องจากระบบนี้ยังไม่แพร่หลายเท่าไรนักสำหรับในประเทศไทยและอาจเป็นพื้นฐานในการพัฒนาเครื่องดนตรีอื่น ๆ โดยเฉพาะดนตรีไทยให้สามารถนำมาประยุกต์เล่นภายในระบบนี้ได้ ซึ่งเป็นระบบที่เป็นมาตรฐานทั่วโลก

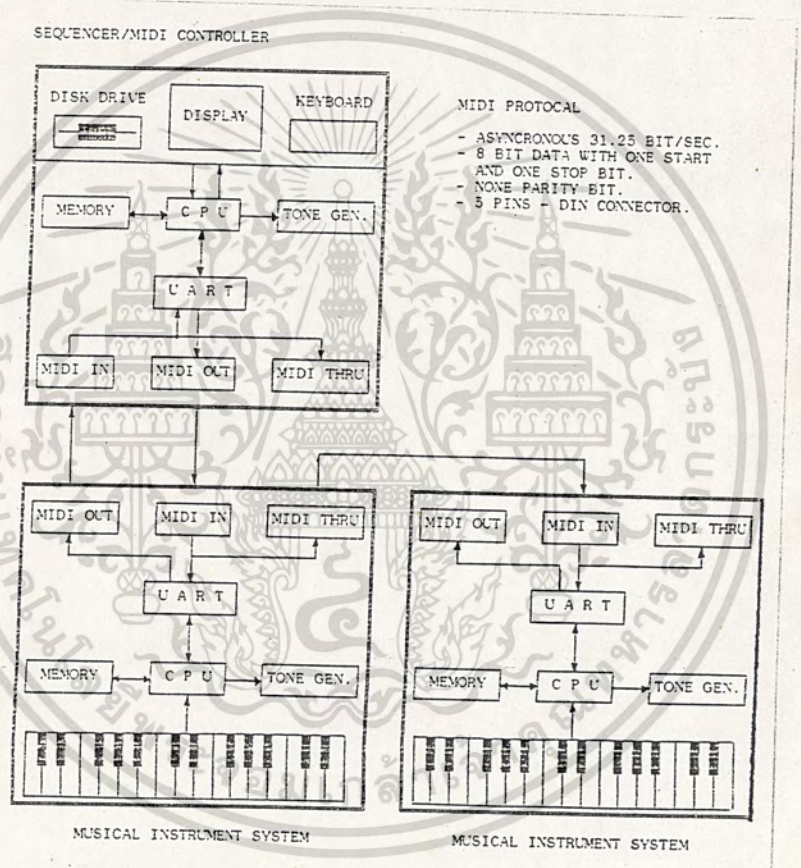
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีของระบบมิดี้

หลักการ และ ข้อกำหนดในระบบมิดี้

หลักการ

มิดี้ (MIDI) ย่อมาจาก Musical Instrument Digital Interface เป็นระบบมาตรฐานสำหรับการสื่อสารข้อมูลระหว่างเครื่องดนตรี โดยจะมีการสื่อสารกันแบบ อซิงโครนัส (asynchronous) ซึ่งเขียนแผนภาพแสดงการทำงานได้ดังนี้



รูปที่ 1 แผนภาพแสดงโครงสร้างของเครื่องดนตรี

จากรูป อุปกรณ์ควบคุม (controlling instrument) เช่น คอมพิวเตอร์ สามารถควบคุมการทำงานของเครื่องดนตรีได้โดยคอมพิวเตอร์จะส่งรหัสมิดี้ไปยังคีย์บอร์ดตัวที่ 1 และใช้ออพติคัลไอโซเลเตอร์ (optoisolator) เป็นตัวคัปปลิ้งสัญญาณส่วน UART จะเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม (และเปลี่ยนจากแบบอนุกรมให้เป็นแบบขนานด้วย)

ข้อมูลที่ได้จะเข้ามายัง ส่วนควบคุมภายในเครื่องดนตรี (micro controller) เพื่อตีความ แล้วสั่งให้คีย์บอร์ดทำงานตามรหัสมิดี้ที่รับได้

ข้อมูลจากออพติคัลไอโซเลเตอร์ส่วนหนึ่งจะผ่านไปยังบัฟเฟอร์แล้วส่งออกไปเรียกว่า มิดี้ธรู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MIDI thru) ซึ่งก็คือการลอกแบบข้อมูลที่ได้รับมานั่นเอง ข้อมูลส่วนนี้จะถูกส่งไปให้คีย์บอร์ด 2 จะเห็นได้ว่า คีย์บอร์ดตัวแรกกับตัวที่สองจะติดต่อกันได้ โดย คีย์บอร์ดตัวแรกจะควบคุมคีย์บอร์ดตัวที่สอง ซึ่งอาศัยข้อมูลที่ได้มาจากคอมพิวเตอร์

ถ้าไม่อาศัยข้อมูลจากคอมพิวเตอร์ เมื่อมีการเล่น คีย์บอร์ด 1 (ใช้คนเล่น) ส่วนควบคุมภายในคีย์บอร์ด 1 จะส่งรหัสมิดี้ออกไป และคอมพิวเตอร์จะได้รับข้อมูลนี้ แล้วนำมาทำขบวนการต่าง ๆ ได้ เช่น เก็บข้อมูลนี้ไว้ในดิสก์ ในลักษณะนี้จะคล้ายกับเทปบันทึกเสียงแต่แทนที่จะเก็บเป็นสัญญาณเสียง คอมพิวเตอร์จะเก็บในรูปของรหัสมิดี้ และเมื่อต้องการเล่นซ้ำ (play back) ทางคอมพิวเตอร์ก็จะส่งรหัสมิดี้นี้กลับไปยังเครื่องดนตรี และเครื่องดนตรีก็จะทำงาน (เล่น) ตามรหัสมิดี้เดิม ซึ่งเป็นการสร้างเสียงขึ้นมาใหม่ (regenerate) นั่นเอง

จะเห็นว่าเมื่อเครื่องดนตรีสามารถสื่อสารกันภายใต้มาตรฐานเดียวกันแล้ว เราสามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง เช่น

1. ควบคุมเครื่องดนตรีได้ทีละหลาย ๆ เครื่อง : เนื่องจากเครื่องดนตรีทุกเครื่องสื่อสารด้วยภาษาเดียวกัน

2. สามารถส่งรหัสมิดี้ไปควบคุมกลองอิเล็คโทรนิค (Drum machine setups) เพื่อให้มีจังหวะตรงกับเครื่องดนตรีอื่น ๆ (synchronized) ได้

3. อุปกรณ์เค่งเสียง (sound effect) ต่าง ๆ สามารถควบคุมได้ด้วยรหัสมิดี้

4. ใช้ในการสร้างเสียง (voice quality) ของเครื่องดนตรีประเภท ซินธิไซเซอร์ (voice editing software)

5. ข้อมูลจากเครื่องดนตรีเครื่องหนึ่งอาจนำไปใช้กับอีกเครื่องได้ (Data transfer)

6. ถ้าใช้การเก็บรหัสมิดี้ แทนที่จะเก็บเป็นสัญญาณเสียง เราสามารถเล่นซ้ำได้ดังที่กล่าวไปแล้ว ซึ่งจะคล้ายกับการบันทึกเสียง (Tepeless recording)

7. ประโยชน์จากการสื่อสาร เช่นสามารถส่งข้อความมิดี้ผ่านโมเด็ม (modem) ได้ ดังนั้นแม้จะอยู่กันคนละประเทศ นักดนตรีก็ยังสามารถส่งดนตรีของเขาไปให้ห้องอัดเสียงได้ทางโทรศัพท์ โดยไม่มีความเพี้ยนของเสียงดนตรีเลย เนื่องจากส่งเป็นรหัส

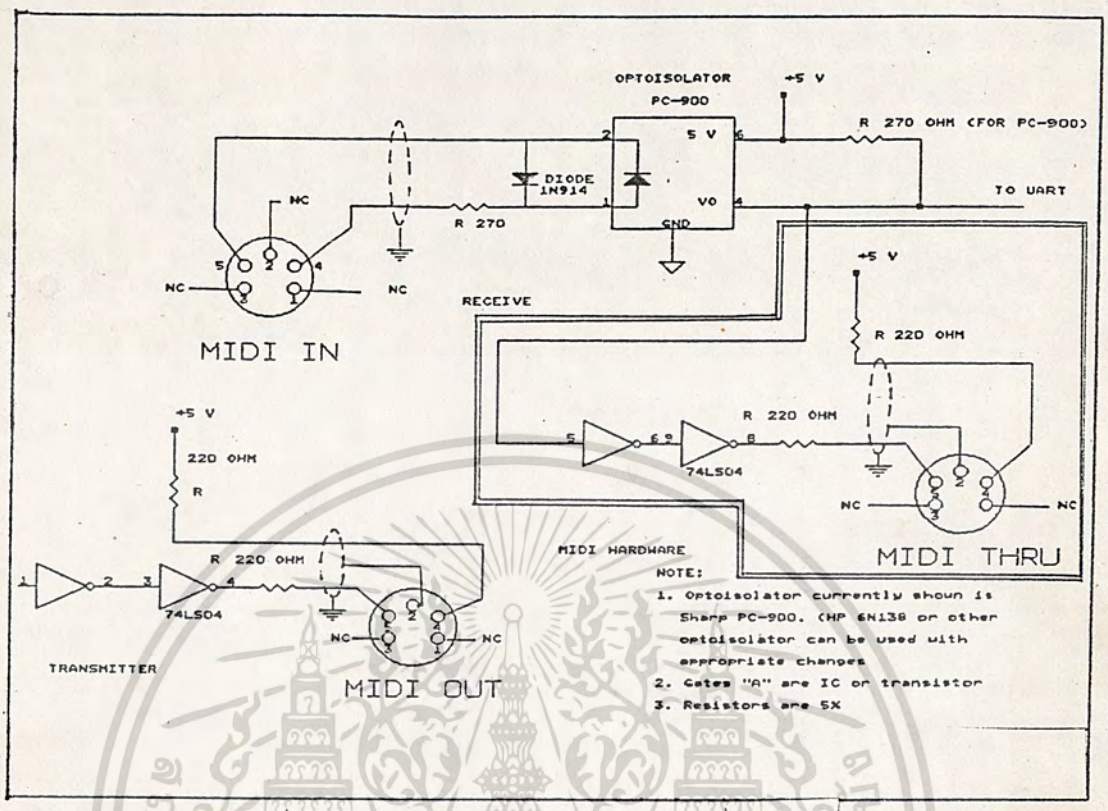
สำหรับเครื่องดนตรีที่ไม่ใช้ไฟฟ้า (acoustic) ก็สามารถใช้ร่วงกับเครื่องดนตรีอื่นด้วยระบบมิดี้ได้ ถ้าเราสามารถทำให้มันส่งข้อมูลออกไปตามข้อกำหนดของระบบมิดี้

<u>ข้อกำหนดของระบบมิดี้</u>

-ข้อกำหนดโดยทั่วไป

มิดี้เป็นมาตรฐานที่จะให้ ซินธิไซเซอร์ (synthesizer) ,ซีเควนเซอร์ (sequencer) คอมพิวเตอร์,อุปกรณ์สร้างจังหวะ ฯลฯ สามารถติดต่อสื่อสารกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 แผนภาพแสดงข้อกำหนดในการเชื่อมต่อ

อุปกรณ์ที่ใช้ระบบมิดี้ มักจะ รับ-ส่ง ข้อความในระบบมิดี้ได้ โดยทางด้านรับจะรับข้อความ (message) ที่เป็นรหัสมิดี้แล้วทำงานตามความหมายของข้อความนั้น อุปกรณ์ทางฮาร์ดแวร์ที่สำคัญประกอบด้วย ออพโต้ไอโซเลเตอร์,Universal Asynchronous Receiver-Transmitter (UART)ส่วนทางด้านส่งก็จะส่งรหัสมิดี้ออกไป โดยมี UART และ ตัวขับสัญญาณ (line driver) เป็นองค์ประกอบสำคัญ

-ฮาร์ดแวร์

สื่อสารแบบอซิงโครนัส โดยใช้ความเร็วในการรับ-ส่งที่ 31.25 kBaud มีบิทเริ่มและบิทหยุด (start bit,stop bit) อย่างละ 1 บิท,บิทข้อมูลมี 8 บิท รวมทั้งหมด 10 บิท ต่อข้อมูล 1 ไบท์ซึ่งใช้เวลาส่ง 320 ไมโครวินาที มีวงจรเป็นดังรูป MIDI HARDWARE

จากรูป กระแสในลูป = 1.5 mA โดยที่ โลจิก 0 จะมีกระแสไหล ทางด้านรับควรมีออพโต้ไอโซเลเตอร์เป็นตัวคัปปลิ้งสัญญาณ ซึ่งเป็นชนิดทำงานที่ความเร็วสูง เช่น Sharp PC-900 เวลาในการสวิทช์ ควรน้อยกว่า 2 ไมโครวินาที

-อุปกรณ์สำหรับเชื่อมต่อ (connector)

เป็นแบบ DIN 5 ขา (แจ็คกลม) ตัวเมีย ขาที่ใช้คือขา 5,4 ส่วนขา 2 ต่อกราวนด์ นอกจากนั้นไม่ใช้ ปล่อยไว้เฉย ๆ

ความยาวของสาย (MIDI cable) ไม่เกิน 15 เมตร หรือ 50 ฟุต และควรมีการพันเกลียว (twist) เพื่อลดสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อของคอนเนคเตอร์ :MIDI IN (ข้อมูลเข้า),MIDI OUT (ข้อมูลออก) และ MIDI THRU ซึ่งเป็นข้อมูลขาออกที่ได้มาจากการก๊อปปี้ข้อมูล มิดี้ อิน โดยตรง

-รูปแบบของข้อมูล (data format)

แต่ละข้อความในระบบมิดี้ มักมีหลายไบท์โดยประกอบด้วยไบท์สถานะ(status byte) ตามด้วยไบท์ข้อมูล (data byte) อีก 1 หรือ 2 ไบท์ ยกเว้นข้อความเกี่ยวกับเวลา (real-time message) และข้อความพิเศษ (exclusive message) ซึ่งจะกล่าวถึงภายหลัง ประเภทของข้อความ จะแบ่งออกเป็น 2 ประเภทใหญ่ ๆ คือ แชนแนล กับ ระบบ

-ข้อความเกี่ยวกับแชนแนล

ในไบท์สถานะจะใช้ 4 บิท เป็นตัวกำหนดแชนแนลของไบท์สถานะ (สามารถกำหนดได้ 16 แชนแนล) ทางด้านรับจะตอบสนอง เฉพาะไบท์สถานะที่มีแชนแนลตรงกับแชนแนลของตนเท่านั้น ถ้าไม่ตรงกันจะไม่สนใจ

ข้อความเกี่ยวกับแชนแนล มี 2 ประเภท

Voice -ใช้ควบคุมเสียง ข้อความเกี่ยวกับเรื่องเสียงจะถูกส่งผ่านทางแชนแนลเสียง (voice channel)

Mode -ใช้กำหนดลักษณะการตอบสนองต่อข้อความเกี่ยวกับเรื่องเสียงของเครื่องดนตรี โดยจะส่งมาทาง แชนแนลพื้นฐาน (Basic channel) ของเครื่องดนตรี

-ข้อความเกี่ยวกับระบบ

ข้อความเกี่ยวกับระบบจะไม่มีการกำหนดแชนแนล ดังนั้นไม่ว่าด้านรับจะเป็นแชนแนลไหน ก็จะตอบสนองต่อข้อความเกี่ยวกับระบบเสมอ

ข้อความเกี่ยวกับระบบมี 3 แบบ

1. ข้อความทั่วไป (common message) เป็นข้อความสำหรับทุกอุปกรณ์ในระบบ
2. ข้อความเกี่ยวกับเวลา จะมีแต่ไบท์สถานะเท่านั้น ไม่มีไบท์ข้อมูล สำหรับข้อความประเภทนี้จะส่งเมื่อไหร่ก็ได้ แม้แต่ส่งคั่นระหว่างไบท์ของข้อความอื่น
3. ข้อความพิเศษ (exclusive message) มีหลายไบท์ข้อมูล และจะจบด้วยไบท์ที่เรียกว่า End Of Exclusive (EOX) หรือไบท์สถานะอื่น ๆ เป็นข้อความที่ใช้เฉพาะกับอุปกรณ์ของผู้ผลิตที่กำหนดไว้ในรหัสประจำเครื่องของแต่ละบริษัทผู้ผลิตหรือ Identification Codes (ID codes) เท่านั้น ถ้าอุปกรณ์ทางด้านรับมีรหัสประจำเครื่องไม่ตรงกับข้อความพิเศษที่ได้ มันจะไม่สนใจไบท์ข้อมูลที่ตามมา ผู้ผลิตแต่ละรายสามารถกำหนดรูปแบบ ความหมายข้อมูลของตนเองได้ โดยส่งเป็นข้อความพิเศษ ที่มีรหัสประจำเครื่องของตน (ซึ่งเป็นรหัสเดียวกันกับทางด้านรับ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ประเภทของข้อมูล (DATA TYPES)

ไบท์สถานะ มี 8 บิท โดยมีบิทที่มีนัยสำคัญสูงสุด (MSB) ถูกเซ็ท (เป็น 1)ไบท์สถานะจะเป็นตัวบอกถึงประเภทของข้อความ และบอกให้รู้ว่าไบท์ข้อมูลที่ตามมาจะเอาไปใช้เพื่ออะไร

สถานะที่กำลังทำงาน (running status) สำหรับข้อความเกี่ยวกับเสียง และโหมดเท่านั้น เมื่อด้านรับได้รับไบท์สถานะและทำขบวนการต่าง ๆ ด้านรับจะยังคงอยู่ในสถานะนั้นจนกว่าจะได้รับไบท์สถานะใหม่ที่ไม่เหมือนเดิม ดังนั้น ถ้ามีการส่งไบท์สถานะเดิมซ้ำมาอีกครั้ง อาจทำเมื่อต้องการแก้ไขไบท์ข้อมูลที่ผิดโดยส่งใหม่ ภายใต้สถานะที่กำลังทำงานข้อความที่สมบูรณ์ต้องประกอบด้วยไบท์ข้อมูลที่ถูกต้องตามข้อกำหนด

สถานะที่กำลังทำงานจะหยุดเมื่อมีไบท์สถานะใหม่เข้ามา เว้นแต่ ถ้ามีข้อความเกี่ยวกับเวลาเข้ามา มันจะยังจงหวะสถานะที่กำลังทำงานแต่เพียงชั่วคราวเท่านั้น

สถานะที่ไม่สามารถทำงานได้ (Unimplement Status) ทางด้านรับจะไม่สนใจไบท์สถานะที่มันไม่สามารถทำงานตามได้ และจะไม่สนใจไบท์ข้อมูลที่ตามมา

สถานะที่ไม่มีการกำหนดความหมาย (Undefined Status)จะไม่มีการใช้ไบท์สถานะที่ไม่มีความหมาย ควรระวังในขณะเปิดเครื่อง หรือ ปิดเครื่อง อาจส่งข้อมูลผิดพลาดได้

ไบท์ข้อมูล จะถูกส่งตามหลังไบท์สถานะ โดยมีไบท์ข้อมูลประมาณ 1 – 2 ไบท์ (ยกเว้นข้อความเกี่ยวกับเวลา) ไบท์ข้อมูลมี 8 บิท โดยบิทที่มีนัยสำคัญสำสุดถูกรีเซ็ท (เป็น 0) จำนวนและขอบเขต (ค่าที่มีได้) ของไบท์ข้อมูล จะเป็นไปตามที่ไบท์สถานะกำหนด ในแต่ละไบท์สถานะจะส่งไบท์ข้อมูลจำนวนที่ถูกต้องตามไปเสมอ
การตอบสนองต่อข้อความจะมีได้ก็ต่อเมื่อด้านรับได้รับไบท์ข้อมูลตามต้องการ แล้วทางด้านรับจะไม่สนใจกับไบท์ข้อมูลที่ตามหลังไบท์สถานะที่มันไม่รู้จัก

-โหมดต่าง ๆ ในแต่ละแชนแนล (channel Mode)

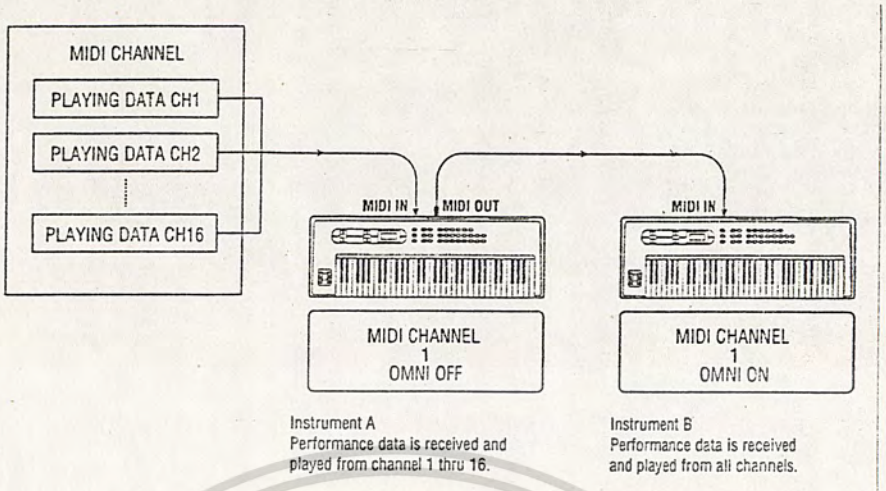
ซินธิไซเซอร์จะประกอบด้วย ส่วนที่สร้างเสียง (sound generator) เสียงที่เกิดขึ้นเรียกว่า voice การกำหนดเสียงเป็นขบวนการเกี่ยวกับข้อมูลพวกการเล่นโน๊ต (note on) กับการเลิกเล่นโน๊ต(note off)จากคีย์บอร์ดซึ่งเสียงที่มีโน๊ตต่าง ๆ จะถูกเล่นได้ในจังหวะที่ถูกต้อง

ข้อความเกี่ยวกับโหมด (mode message)ใช้กำหนดลักษณะการสร้างเสียงของเครื่องดนตรีประเภทซินธิไซเซอร์ ได้แก่ Omni (On/off),Poly และ Mono

Mono on จะเป็นการกำหนดให้เสียงในแชนแนลเสียงมีได้ 1 เสียง เท่านั้นในเวลาเดียวกัน (Mono phonic)

Mono off (Poly on) : ซินธิไซเซอร์ที่อยู่ในโหมดนี้สามารถสร้างเสียงได้พร้อม ๆ กันทีละหลาย ๆ เสียง (Poly phonic) ซึ่งทั้งนี้จะขึ้นอยู่กับความสามารถในการสร้างเสียงของตัวซินธิไซเซอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 แผนภาพแสดงการทำงานในโหมดต่าง ๆ

สำหรับ Omni ถ้า Omni on ตัวรับจะสามารถรับข้อความเกี่ยวกับเสียงได้จากทุกแชนแนลเสียงโดยไม่มีความต่างกันของแชนแนล ถ้า Omni offด้านรับจะรับเฉพาะข้อความเกี่ยวกับเสียงที่มีแชนแนลตรงกับของตนเท่านั้น

ทางด้านรับที่ถูกกำหนดเป็นแชนแนลพื้นฐานจะมีลักษณะการทำงานได้ 4 โหมดดังนี้

โหมด 1. Omni On , Poly

โหมด 2. Omni On , Mono

โหมด 3. Omni off , poly

โหมด 4. Omni off , Mono

ทางด้านส่งและรับจะทำงานได้ทีละ 1 โหมดเท่านั้นในเวลาเดียวกันและตามปกติจะส่งและรับในโหมดเดียวกัน

ด้านรับจะรับรู้ข้อความเกี่ยวกับโหมดทางแชนแนลพื้นฐานเท่านั้น ส่วนข้อความเกี่ยวกับเสียงอาจได้รับทางแชนแนลพื้นฐานหรือแชนแนลอื่น ๆ ก็ได้ซึ่งเรียกรวมกันว่าแชนแนลเสียง

โดยทั่วไปมักกำหนดให้แชนแนลแรก (0) เป็นแชนแนลพื้นฐาน

- ความหมายของข้อความต่าง ๆ ในระบบมิดี้

1. ข้อความที่เกี่ยวกับเสียงในแต่ละแชนแนล (channel voice message)

การเล่นโน๊ต (note on)

เป็นข้อความที่บอกให้รู้ถึงการเล่นโน๊ต (เช่นการเอานิ้วกดไปที่คีย์บอร์ด) มี 3 ไบท์

1. ไบท์สถานะคือ 1001 nnnn (ฐานสอง) โดยที่ nnnn คือหมายเลขแชนแนล (0 - 15)
2. ไบท์ข้อมูลแรกจะบอกให้รู้ว่าโน๊ตไหนที่ถูกกด (note number) มีค่าได้ตั้งแต่ 0 - 127 โดย 0 จะเป็นโน๊ตที่ต่ำที่สุด 127 เป็นโน๊ตที่สูงที่สุด แต่โดยทั่วไปเครื่องดนตรีจะรับรู้โน๊ตได้น้อยกว่า เช่น ซินธิไซเซอร์ที่มี 5 ออคเตฟ จะรับรู้หมายเลขโน๊ตได้ตั้งแต่ 36 - 96 หมายเลขโน๊ตของโดกลาง (middle C) คือ 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ไบท์ข้อมูลต่อมาจะบอกถึงน้ำหนักในการเล่นหรือความเร็วที่กดคีย์ (key velocity) เหมือนกับการเล่นเปียโนถ้ากดคีย์เปียโนด้วยความเร็วจะได้เสียงที่ดัง และ ถ้ากดช้าก็จะได้เสียงที่เบาเช่นกัน ความเร็วของคีย์มีค่าได้ตั้งแต่ 0 - 127 โดยความเร็ว = 127 จะเป็นเสียงที่ดังที่สุด ส่วนความเร็ว = 0 จะไม่มีเสียงออกมา (note off) ซึ่งมีความหมายเหมือนการยกนิ้วขึ้นจากคีย์สำหรับเครื่องคนตรีที่ไม่มีการตอบสนองต่อความเร็วของคีย์จะกำหนดให้ข้อมูลส่วนนี้มีค่าเท่ากับ 64

การเลิกเล่นโน๊ต (note off)

เป็นข้อความที่บอกให้รู้ถึงการเลิกสร้างเสียง (เช่นเกิดขึ้นในขณะที่ยกนิ้วขึ้นจากคีย์) มี 3 ไบท์

1. ไบท์สถานะ คือ 1000nnnn (ฐานสอง) โดยที่ nnnn คือ หมายเลขแชนแนล
2. ไบท์ข้อมูลแรกบอกให้รู้ว่าโน๊ตตัวไหนที่จะให้เลิกสร้างเสียง (key number)
3. ความเร็วที่ปล่อยคีย์ (key off (release) velocity) เป็นไบท์ข้อมูลที่บอกถึงความเร็วในการปล่อยคีย์ ในกรณีที่อุปกรณ์ไม่มีความเร็วของคีย์จะให้ไบท์นี้มีค่าเป็น 64

เมื่อเครื่องคนตรีได้รับข้อความเกี่ยวกับการเล่นโน๊ต มันจะผลิตเสียงค้างอยู่อย่างนั้นจนกว่าจะได้รับข้อความเกี่ยวกับการเลิกเล่นโน๊ตของโน๊ตตัวนั้นมันถึงจะหยุดทำเสียง ดังนั้น เมื่อมีการส่งข้อความเกี่ยวกับการเล่นโน๊ตออกไปแล้ว จะส่งข้อความเกี่ยวกับการเลิกเล่นโน๊ตตามมาด้วยทุกครั้ง มิฉะนั้นจะเกิดเสียงค้าง

การกำหนดความดันของคีย์แบบโพลีโฟนิค (Polyphonic Key pressure)

คีย์บอร์ดในระบบมิดี้บางเครื่องไม่เพียงแต่ตอบสนองความเร็วของคีย์ได้เท่านั้นยังตอบสนองน้ำหนัก (ความดัน) ของนิ้วที่กดลงไปอีกหลังจากที่กดคีย์ลงไปสุดแล้ว มี 3 ไบท์

1. ไบท์สถานะ คือ 1010nnnn
2. หมายเลขของโน๊ต
3. ความดันของคีย์ (key pressure value) มีค่าตั้งแต่ 0-127 โดย 0 หมายถึง ไม่มีความดันเลย และ 127 จะมีความดันสูงสุด

ประโยชน์ของโพลีโฟนิคคีย์เพรชเชอร์ คือจะให้ความรู้สึกของเครื่องเป่า เช่น ทรัมเป็ต เมื่อตั้งเสียงของซินธิไซเซอร์เป็นเสียงทรัมเป็ต การกดคีย์ลงไปอีกหลังจากสุดคีย์แล้วจะมีลักษณะเหมือนการที่เราเป่าลมเข้าไปในทรัมเป็ตอีก

คีย์บอร์ดมีโพลิโฟนิคคีย์เพรชเชอร์จะมีระบบที่ซับซ้อน และมีราคาแพงมาก ซึ่งจะหาคีย์บอร์ดที่มีระบบนี้ได้น้อยราย

การกำหนดความดันรวม (Overall Pressure)

ต่างกับโพลิโฟนิคคีย์เพรชเชอร์ตรงที่ไม่สามารถบอกความดันแยกกันของแต่ละคีย์ที่ถูกกดได้แต่จะบอกเป็นความดันเฉลี่ยของทุก ๆ คีย์ในเวลานั้น ๆ มีอยู่ด้วยกัน 2 ไบท์

1. ไบท์ข้อมูลคือ 1101nnnn (nnnn คือ หมายเลขแชนแนล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ความดันของแชนแนล (channel pressure value) มีค่าตั้งแต่ 0 - 127
โดยที่ 0 จะไม่มีความดัน และ 127 มี ความดันมากที่สุด

ข้อมูลเกี่ยวกับการควบคุม (Control Change)

ใช้ในการปรับแต่ง เสียง สำหรับข้อมูลเกี่ยวกับการควบคุมในปัจจุบันยังไม่มีมาตรฐานที่แน่นอน ระหว่างแต่ละบริษัทผู้ผลิตมี 3 ไบท์

1. ไบท์สถานะ คือ 1011nnnn
2. หมายเลขตัวควบคุม (controller number) มีค่าตั้งแต่ 0 - 127 ดูความหมายได้จาก MIDI Implementation Sheet ของแต่ละบริษัทผู้ผลิต
3. ค่าที่จะให้แก่ตัวควบคุม (controller value)

- การเอื้อนเสียง (Pitch Bend)

เป็นลักษณะของการเอื้อนเสียงจากสูงไปต่ำหรือจากต่ำไปสูงโดยที่ไม่ต้องเปลี่ยนตำแหน่งของคีย์ที่กด การเอื้อนเสียงมี 2 ไบท์

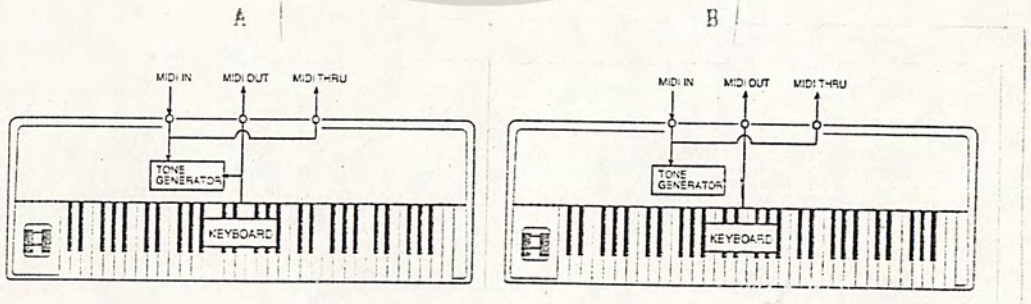
1. ไบท์สถานะ คือ 1110nnnn
2. ค่าที่จะให้เอื้อน (pitch bend value)

ข้อมูลเกี่ยวกับโปรแกรมเสียง (Progra Change) หรือบางทีเรียกว่า Program Select จะบอกให้รู้ถึงการใช้โปรแกรมเสียงต่าง ๆ ในเครื่องดนตรี มี 2 ไบท์

1. ไบท์สถานะคือ 1100nnnn
2. โปรแกรมเสียงที่ใช้ (selected program number) มีค่าตั้งแต่ 0 ถึง 127

<u>ข้อมูลเกี่ยวกับลักษณะการทำงานในแต่แชนแนล</u> (Channel Mode Messages)

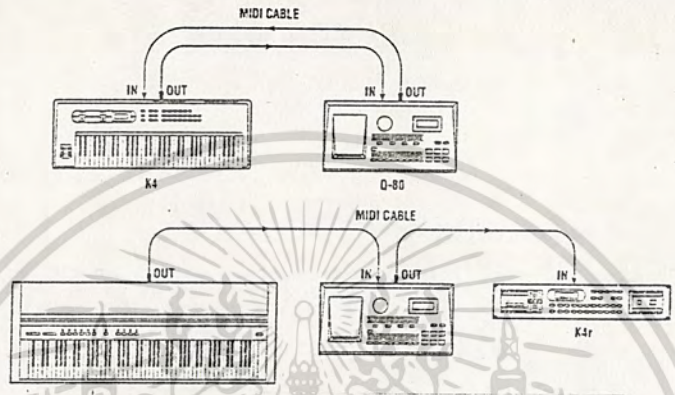
เป็นข้อความเกี่ยวกับโหมดในการทำงาน เป็นส่วนหนึ่งของข้อความเกี่ยวกับการควบคุมการควบคุมคีย์บอร์ดแบบ โลคอล/รีโมต (Local/Remote Keyboard Control) ใช้ในการต่างเครื่องดนตรี เข้าด้วยกันโดยควบคุมจากเครื่องเดียว



รูปที่ 4 แผนภาพแสดงทางเดินของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป (a) ถ้าเป็นการควบคุมแบบโลคอล (Local Control On) ข้อมูลทางด้านเสียง (เช่น การเล่นโน๊ต) จากคีย์บอร์ดจะเข้าไปยังส่วนสร้างเสียงและข้อมูลเดียวกันนี้จะออกไปทางมิดี้เอ้าท์ด้วย (ข้อมูลจากมิดี้เอ้าท์ใช้ส่งให้กับอุปกรณ์อื่น) ดังนั้นในการพ่วงซินธิไซเซอร์เข้าด้วยกันดังรูปที่แสดงการเชื่อมต่อ เมื่อเล่น ซินธิไซเซอร์ #1 จะมีเสียงออกมาจากทั้ง ซินธิไซเซอร์ #1 และ ซินธิไซเซอร์ #2 (ซินธิไซเซอร์ #2 เล่นตามข้อมูลที่ส่งมาจาก ซินธิไซเซอร์ #1)



รูปที่ 5 แผนภาพแสดงการเชื่อมต่อของอุปกรณ์มิดี้

ในกรณีที่เป็น การควบคุมแบบรีโมด ((Local control off) (รูป(b)) ข้อมูลจากคีย์บอร์ดจะส่งไปที่ มิดี้เอ้าท์โดยตรง (ไม่ส่งให้ส่วนสร้างเสียง) ดังนั้นในรูปแสดงการเชื่อมต่อถ้าอยู่ในลักษณะการควบคุมแบบรีโมด เมื่อเล่นที่ซินธิไซเซอร์#1 จะมีเสียงออกมาจาก ซินธิไซเซอร์#2 เท่านั้น

- ข้อมูลเกี่ยวกับการควบคุมคีย์บอร์ดแบบ โลคอล/รีโมด มี 3 ไบท์

1. ไบท์สถานะ คือ 1011nnnn (ซึ่งเป็นไบท์สถานะของข้อความเกี่ยวกับการควบคุม) nnnn คือ แชนแนลพื้นฐาน
2. หมายเลขตัวควบคุม = 122
3. ถ้าเป็นแบบรีโมดจะให้ไบท์นี้เป็น 0, ถ้าเป็นแบบโลคอลจะเป็น 127

การเลิกเล่นโน๊ตทุกตัว (All Note Off)

เป็นการหยุดการสร้างเสียง (หยุดทั้งแชนแนลโดยไม่สนว่าเป็นโน๊ตตัวไหน) มี 3 ไบท์

1. ไบท์สถานะคือ 1011nnnn (เป็นไบท์สถานะของข้อความเกี่ยวกับการควบคุมนั่นเอง nnnn คือ แชนแนบพื้นฐาน
2. หมายเลขตัวควบคุม = 123
3. เป็นไบท์ที่ไม่มีความหมายอะไร ใส่ไบท์ให้ครบ 3 ไบท์เท่านั้น (dummy byte)

เนื่องจากข้อความเกี่ยวกับการควบคุมมี ความยาว 3 ไบท์

- การเลือกโหมด Omni/Poly/Mono

ได้อธิบายเกี่ยวกับโหมดเหล่านี้ไปแล้วในตอนต้น ทุกครั้งที่เปลี่ยนโหมดจะเลิกเล่นโน๊ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028789

ทุกตัว รูปแบบของข้อความนี้ก็คล้าย ๆ กับข้อความเกี่ยวกับการควบคุมอื่น ๆ ประกอบด้วย

1. ไบท์สถานะคือ 1011nnnn (nnnn เป็นแชนแนลพื้นฐาน)
2. หมายเลขตัวควบคุมของแต่ละโหมด เป็นดังนี้

   Omni Off = 124

   Omni On = 125

   Mono On = 126

   Poly On = 127

3. ไบท์ที่ 3 จะให้เป็น 0 สำหรับ ตัวควบคุมหมายเลข 123,124,125, 127 ในกรณีของโหมดโมโน (126) ไบท์จะเป็นการกำหนดแชนแนลที่จะให้เป็นโมโน

ข้อมูลทั่วไปของระบบ (System Common Message) (สำหรับทุกแชนแนล)

เนื่องจากข้อมูลประเภทนี้เป็นข้อมูลของระบบ ดังนั้นจะไม่มีการกำหนดแชนแนล นั่นคือทุกแชนแนลต้องตอบสนองต่อข้อมูลทั่วไปของระบบ

การชี้ตำแหน่งของเพลง (Song Position Pointer)

บอกให้รู้ถึงตำแหน่งของเพลงที่กำลังเล่นอยู่ การบอกตำแหน่งจะบอกว่าเพลงได้ดำเนินไปแล้วกี่บีท (MIDI beat)บอกได้สูงสุด 16,384 บีทโดย 1 บีทมีค่าเท่ากับโน๊ตซิกเจ็ดสองชั้นหนึ่งตัว

การเลือกเพลง (Song Select)

คล้าย ๆ กับการเลือกโปรแกรมเสียง แต่แทนที่จะเป็นโปรแกรมเสียงจะเป็นโปรแกรมรูปแบบ (pattern) สั้น ๆ ของเพลงแทน (ความยาวไม่กี่ห้อง)โดยมากจะใช้ในกลองอิเลคโทรนิค แต่ละเพลงก็คือแต่ละจังหวะ เช่น เพลงที่หนึ่ง (song 1) เป็น จังหวะร็อค และ เพลงที่สอง (song 2) เป็นจังหวะวอลซ์

ข้อความพิเศษของระบบ (System Exclusive) : ได้อธิบายแล้วในตอนต้น

ข้อความเกี่ยวกับเวลาของระบบ (System Real Time Message) (ทุกแชนแนล)

เป็นข้อความที่ทำให้เครื่องดนตรีต่าง ๆ เล่นอยู่ในจังหวะเดียวกัน (Synchronized)เช่นในการห่วงซินธิไซเซอร์ กับ กลองอิเล็คโทรนิคเข้าด้วยกันจำเป็นอย่างยิ่งที่อุปกรณ์ทั้งสองนี้จะต้องทำงานอยู่ในจังหวะเดียวกัน ไม่เช่นนั้นก็คงจะฟังไม่รู้เรื่องถ้าปล่อยให้ต่างฝ่ายต่างเล่น

- การรีเซ็ทระบบ (System Reset)

เป็นข้อความที่บอกให้อุปกรณ์ระบบมีดีทุกชิ้นทำการรีเซ็ทตัวเอง(กลับไปอยู่ในสภาพเริ่มแรกหลังจากเปิดไฟเข้าเครื่อง)

- สัญญาณนาฬิกา (Timing Clock)

ในการกำหนดจังหวะจะมีการส่งสัญญาณนาฬิกาออกไปด้วยอัตรา 24 ครั้ง ต่อ ร็อคตัวดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งตัว (ส่งออกไปอย่างต่อเนื่อง) เพื่อให้เป็นจังหวะอ้างอิงในระบบควรใช้สัญญาณนาฬิกาเดียวกัน (มีอุปกรณ์ส่งสัญญาณนาฬิกาเพียงตัวเดียว)

การกลับไปที่จุดเริ่มต้น (Start From First Measure)

เป็นข้อความที่บอกให้ซีเควนเซอร์ และกลองอิเลคโทรนิคกลับไปที่จุดเริ่มต้นของเพลง แล้วเริ่มเล่นเพลงในทันทีที่ได้รับสัญญาณนาฬิกา นั่นคือจะบอกให้แต่ละอุปกรณ์เริ่มเล่นพร้อม ๆ กัน

การหยุด (Stop)

ข้อความนี้จะบอกให้หยุดการกระทำทุกอย่างที่มีผลมาจากการกำหนดจังหวะ (timing-sensitive)

การเริ่มเล่นต่อ (Continue Start)

บอกให้อุปกรณ์ทำการเล่นต่อหลังตำแหน่งเพลงที่ถูกหยุดครั้งสุดท้าย

การตรวจสอบสภาวะการทำงาน (Active Sensing)

เป็นข้อความที่ใช้แก้ปัญหาตัวส่งกับตัวรับถูกตัดขาดออกจากกัน (เช่นสายหลุด) ซึ่งจะก่อให้เกิดความผิดพลาด เช่น ส่งข้อความให้เล่นโน๊ตออกไปแล้วเกิดสายหลุด เมื่อทางด้านส่งส่งข้อความให้เลิกเล่นโน๊ตตามไป ก็จะไปไม่ถึงด้านรับ ทำให้เกิดเสียงค้างที่เครื่องดนตรีด้านรับ ปัญหานี้จะเกิดความเสียหายมากในการแสดงสด

- แอคทีฟเซ็นซิ่ง จะทำงานโดยการส่งข้อความที่เรียกว่า แอคทีฟเซ็นซิ่ง ออกไปเรื่อย ๆ เมื่อด้านรับได้รับ แอคทีฟเซ็นซิ่ง แล้ว ถ้าภายในเวลาสั้น ๆ ที่กำหนดไม่ได้รับข้อความนี้เข้ามาอีกมันจะหยุดการสร้างเสียง

ที่กล่าวมาแล้วทั้งหมดเป็นความหมายของข้อความในระบบมีดี้ต่าง ๆ แต่ โดยทั่วไปเครื่องดนตรีแต่ละชิ้นจะไม่สามารถตอบสนองต่อข้อความเหล่านี้ได้ทุกข้อความ ซึ่งความสามารถในการตอบสนองต่อข้อความต่าง ๆ ของแต่ละอุปกรณ์ จะหาดูได้ใน มิดี้ อิมพลีเมนท์เตชั่น ชีต (Midi Implementation Sheet)ของอุปกรณ์นั้น ๆ สำหรับโครงงานนี้ได้อุปกรณ์มิดี้ (MIDI)ของบริษัท YAMAHA รุ่น PSR-47,รุ่น PSS_680 ซึ่งมี อิมพลีเมนท์เตชั่น ชีต (Implementation Sheet) ดังที่แสดงไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>รายละเอียดเกี่ยวกับโครงงาน</u>

โครงงานนี้จะทำให้ Microcomputer ทำงานควบคุมเครื่องดนตรีต่าง ๆ ให้เล่นเพลงตามต้องการ โดยการบันทึกการเล่นของเครื่องดนตรีแต่ละชิ้นเข้าไปก่อน แล้วจึงส่งข้อมูลทั้งหมดให้เครื่องดนตรีเล่นเป็นเพลงพร้อมกัน โดยในขณะที่มีการรับข้อมูลเข้าไป Computer สามารถส่งข้อมูลเก่าที่เก็บไว้แล้วออกไปพร้อมกันด้วย รวมทั้งการปรับปรุงแก้ไข ข้อมูลดังกล่าวในขณะที่บันทึกด้วย (เช่น การเลื่อนระดับเสียง(TRANSPOSE),การกำหนดระดับความดัง(SET VOLUME) , การกำหนดเสียง(SET VOICE),การกำหนด แทรค(SET TRACK) โดยจะส่งไปให้ แทรค ที่ต้องการ โดยที่ในขณะรับไม่จำเป็นตรงกับ แทรค นั้น)

นอกจากนั้น ไมโครคอมพิวเตอร์ ยังทำหน้าที่เป็น กลองอิเลคทรอนิคส์ (DRUM MACHINE) ด้วยซึ่งผู้ใช้สามารถโปรแกรมฝ่านมอนิเตอร์(MONITOR) ได้อย่างสะดวก

การแก้ไขข้อมูลที่บันทึกเข้าไปก็สามารถกระทำได้โดยสะดวกด้วยวิธีการแสดงผล(DISPLAY)เป็นรูปการกดคีย์ หรือการเล่นในขณะนั้น ๆ ,การลบ,การ COPY และเลื่อนระดับเสียงด้วยเช่นกัน นอกจากนี้ยังมีส่วน ควอนไตซ์(QUANTIZE) ซึ่งจะทำหน้าที่วิเคราะห์ข้อมูลที่เล่นจากนักดนตรีว่าควรจะมีการลงจังหวะอย่างไรถึงจะถูกต้อง ช่วยให้ประสิทธิภาพการบันทึกสมบูรณ์ยิ่งขึ้น

โดยสรุปก็คือ ไมโครคอมพิวเตอร์ จะต้องทำหน้าที่รับใช้เครื่องดนตรีและนักดนตรีอย่างมีประสิทธิภาพที่สุด

<u>ฮาร์ดแวร์ ของ มิดี้ คาร์ด</u>

หลักการทำงานที่สำคัญ มิดี้ คาร์ด (MIDI CARD)อยู่ที่ IC Z80B-SIO ซึ่งจะทำหน้าที่เป็น SERIAL INPUT OUTPUT โดยเราจะทำการโปรแกรมให้ SIO ทำงานใน โหมด อินเทอร์รัพท์ Interrupt Mode ซึ่งมีวงจรตามภาพหน้าถัดไป

Z80B-SIO จะรับความถี่ในการมอด บน คริสตอล(CRYSTAL) ขนาด 2 MHz ซึ่งต่อร่วมอยู่กับ IC 74LS00 ในการส่งข้อมูลจะมี IC 74LS04 ทำหน้าที่เป็น บัฟเฟอร์(BUFFER) โดยจะมีตัวต้านทาน(RESISTOR) ขนาด 220 โอห์ม ซึ่งจะแมทช์(Match) กับ ค่าอิมพีแดนซ์(Impedance) ของ มิดี้ อิน พอดี(ค่ามาตรฐานของ มิดี้)

ในการอ้างพอร์ท(PORT) ของ IBM ถึง Z80B-SIO จะใช้ IC 74LS138 2 ตัว ร่วมกันกับ แนนด์เกท (nand gate) ของ 74LS00 โดยจะต้องสร้าง IORQ ให้กับ Z80B-SIO เนื่องจากการจัดขาของ Z80 ไม่ตรงกับ พอร์ท ของ IBM

สำหรับในโครงงานนี้จะเลือกใช้ CHANNEL B ของ Z80B-SIO เนื่องจากมันมีประสิทธิภาพที่เหนือกว่า CHANNEL A (แต่ในโครงงานนี้ยังไม่ได้ใช้) และ ใช้หมายเลขของ คอนโทรล พอร์ท (CONTROL PORT) คือ 2B1H และ ดาต้า พอร์ท (DATA PORT) คือ 2B0H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80B-SIO จะถูกโปรแกรมให้รับส่งข้อมูลด้วย BAUD RATE FACTOR = 64 ซึ่งจะได้ความถี่ในการมอดดูเลชั่น = 2 MHz/64 = 31.25 KBaud พอดี และมีขนาดข้อมูล 8 Bitพร้อมบิทเริ่ม/บิทหยุด (STOP/START BIT) อย่างละ 1 ตัว

ส่วนประกอบสำคัญอีกส่วนหนึ่ง คือ 8253 ซึ่งเป็น PROGRAMMABLE INTERVAL TIMER โดยจะนำมาใช้ในการทำ ฐานเวลา ให้กับระบบสำหรับตัว 8253 เองจะมี วงจรนับ(COUNTER) อยู่ 3 ตัวอิสระ คือ C0,C1 และ C2 ที่ใช้ใน มิดี้ คาร์ด นี้ เราจะใช้ C0 และ C1 โดยเราจะกำหนดให้ C1 ทำงานในโหมด 3 (MODE 3)ผลิตคลื่นสี่เหลี่ยม(SQUARE WAVE GENERATOR) ซึ่งในโหมดนี้ เราจะใช้ในการนำมาหารความถี่ลงมาจากความถี่ อินพุท ที่ป้อนให้ที่ขา C1 คือ 2 MHz ลงมาเหลือ 1 MHz โดยทำการหาร 2 นั่นเองจากนั้นเราจะนำความถี่ที่หารแล้วจาก C1 มาป้อนเป็น คล็อค(CLOCK) ให้กับ C0 ซึ่งที่ C0 นี้ ก็จะทำงานใน โหมด 3 เช่นกันแต่ค่าที่ใช้ในการหารความถี่ตรงนี้จะเป็นค่าที่ อินพุทมาจาก ผู้ใช้(USER) เพื่อใช้กำหนดให้เป็นฐานเวลาของความเร็วช้าของจังหวะ(TEMPO) จากนั้นนำค่าความถี่จาก C0 ที่ขา OUT 0 ไปหริกที่ IORQ 3 ส่วน Z80B-SIO ที่ขา อินเทอร์รัพท์ (INT)จะต่อกับ IORQ2 เพื่ออินเทอร์รัพท์ซึ่งจะมีการทำงานเมื่อมีการรับส่งข้อมูลจากเครื่องดนตรี

คอนเนคเตอร์(Connector) ในส่วนที่ติดต่อกับอุปกรณ์ มิดี้ จะใช้ แจ็ค ดิน ตามมาตรฐานมิดี้ แต่ในส่วนที่ติดต่อกัน เครื่อง IBM จะใช้ DB CONNECTOR 9 ขา แทนเนื่องจากยังอยู่ในช่วงการศึกษาและพัฒนา

<u>ซอฟท์แวร์</u>

ซอฟท์แวร์ (SOFTWARE) ของโครงงานนี้ใช้ ภาษา ASSEMBLY ในการพัฒนาระบบ เนื่องจากจะต้องเกี่ยวข้องกับ ฮาร์ดแวร์ (HARDWARE) เป็นอย่างมาก ดังจะได้กล่าวต่อไป แต่โดยทั่วไปโปรแกรมจะประกอบไปด้วย 3 ส่วนใหญ่ คือ

1.ส่วนที่ทำหน้าที่เป็น ซีเควนเซอร์(SEQUENCER)ซึ่งจะบันทึกข้อมูลจากอุปกรณ์ มิดี้(MIDI) มาเก็บ และส่งข้อมูลใน หน่วยความจำ ออกไปด้วย

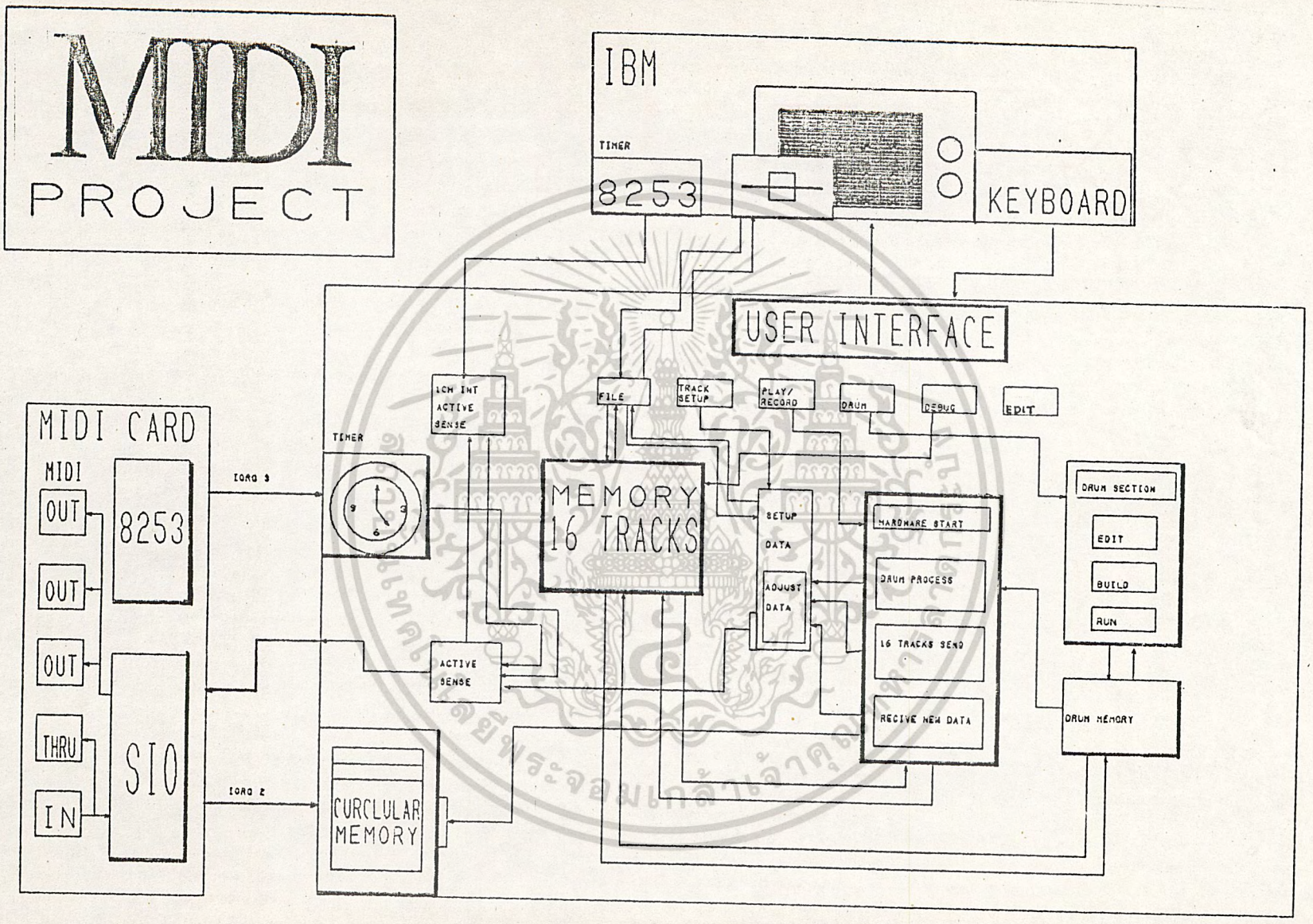
2.ส่วนที่ทำหน้าที่เป็น กลองอิเลคทรอนิคส์ (DRUM MACHINE) ทำหน้าที่ในการตีกลองและให้จังหวะ แก่ระบบในขณะบันทึก เพลง

3.ส่วนปรับปรุงและแก้ไขข้อมูล ซึ่งจะช่วยให้นักดนตรีสามารถทำงานได้อย่างมีประสิทธิภาพ ซึ่งนับได้ว่าเป็นส่วนที่ยาก และสำคัญที่เดียว ในส่วนนี้จะใช้ TURBO PASCAL ในการพัฒนา

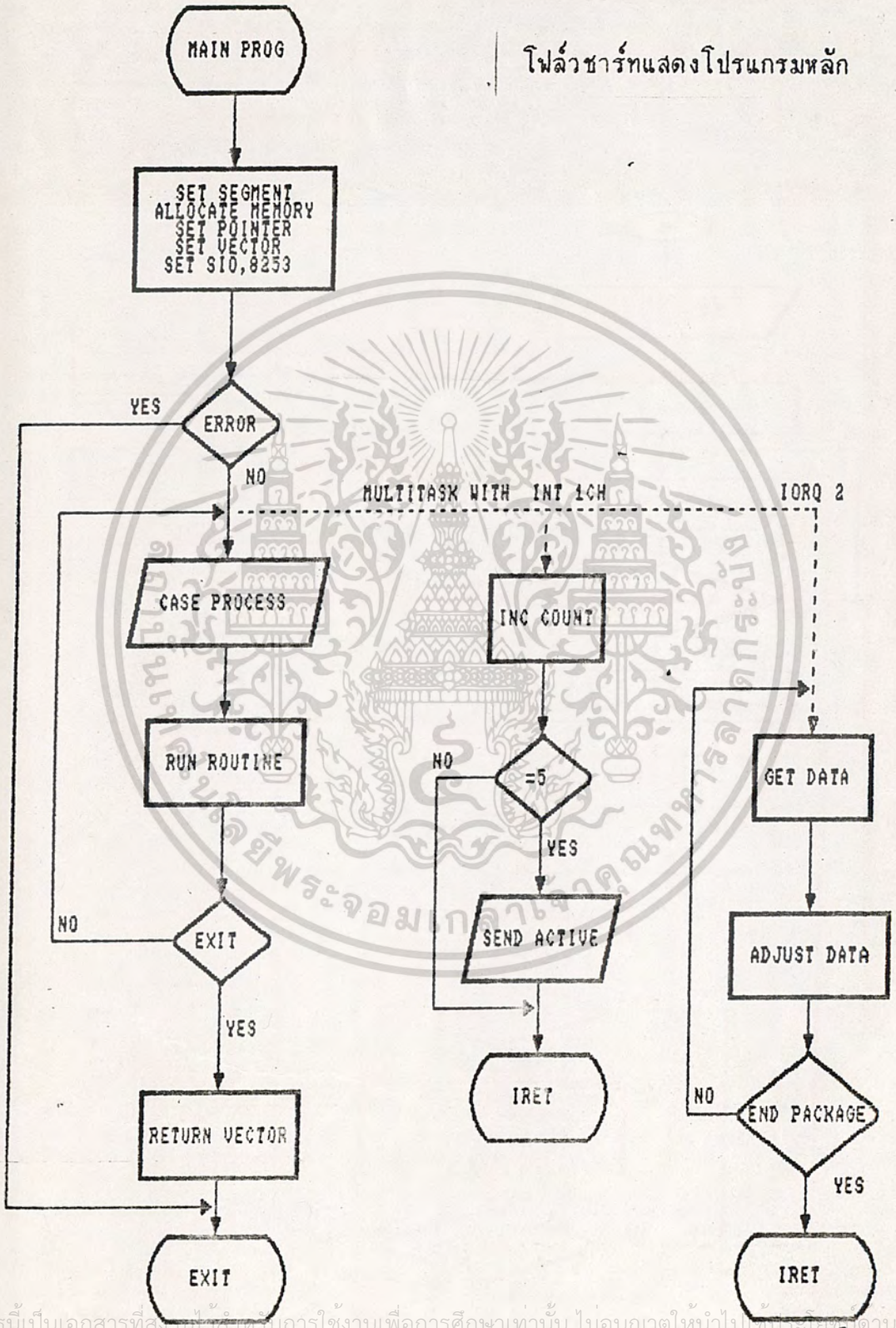
<u>ส่วน SEQUENCER</u>

จะทำการส่งข้อมูลจากที่เก็บไว้ใน หน่วยความจำ ออกไป และรับข้อมูลจาก เครื่องดนตรีแล้วส่งกลับไป (ECHO) และการส่งของข้อมูลแต่ละแทรค จะมี การดัดแปลงข้อมูลต่าง ๆ ตามค่าที่ SETUP ไว้โดยผู้ใช้ ซึ่งการรับข้อมูลนี้จะทำการบันทึกหรือไม่ก็ขึ้นกับว่าเป็นการ เล่น(PLAY)หรือเป็นการ บันทึก (RECORD) ในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

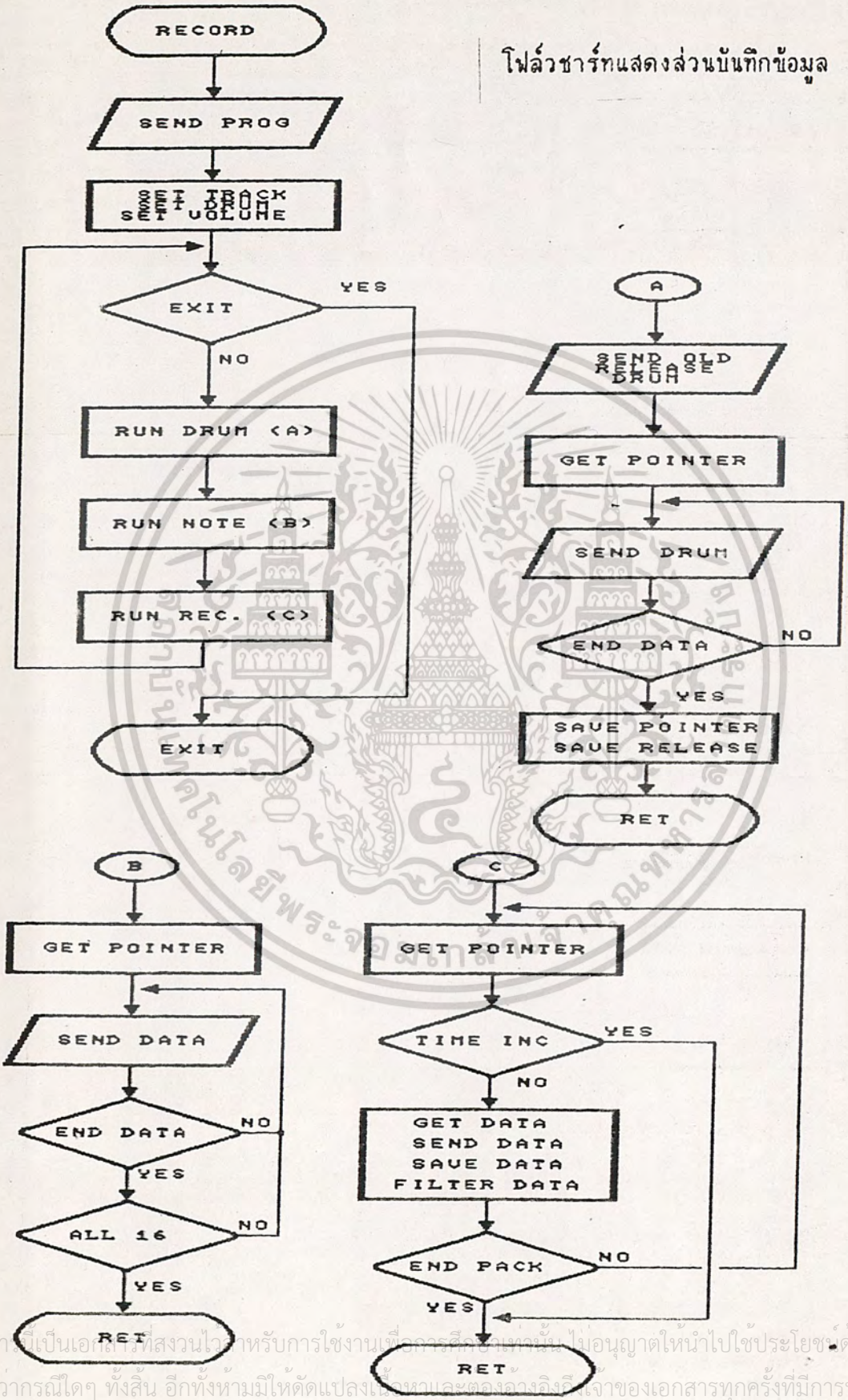

MIDI
PROJECT
IBM
TIMER
8253
KEYBOARD
USER INTERFACE
1CH INT
ACTIVE
SENSE
FILE
TRACK
SETUP
PLAY/
RECORD
DRUM
DEBUG
EDIT
MIDI CARD
MIDI
OUT
OUT
OUT
THRU
IN
8253
SIO
IORQ 3
IORQ 2
TIMER
MEMORY
16 TRACKS
SETUP
DATA
ADJUST
DATA
ACTIVE
SENSE
CURCLULAR
MEMORY
HARDWARE START
DRUM PROCESS
16 TRACKS SEND
RECIVE NEW DATA
DRUM SECTION
EDIT
BUILD
RUN
DRUM MEMORY

โฟล์วชาร์ทแสดงโปรแกรมหลัก



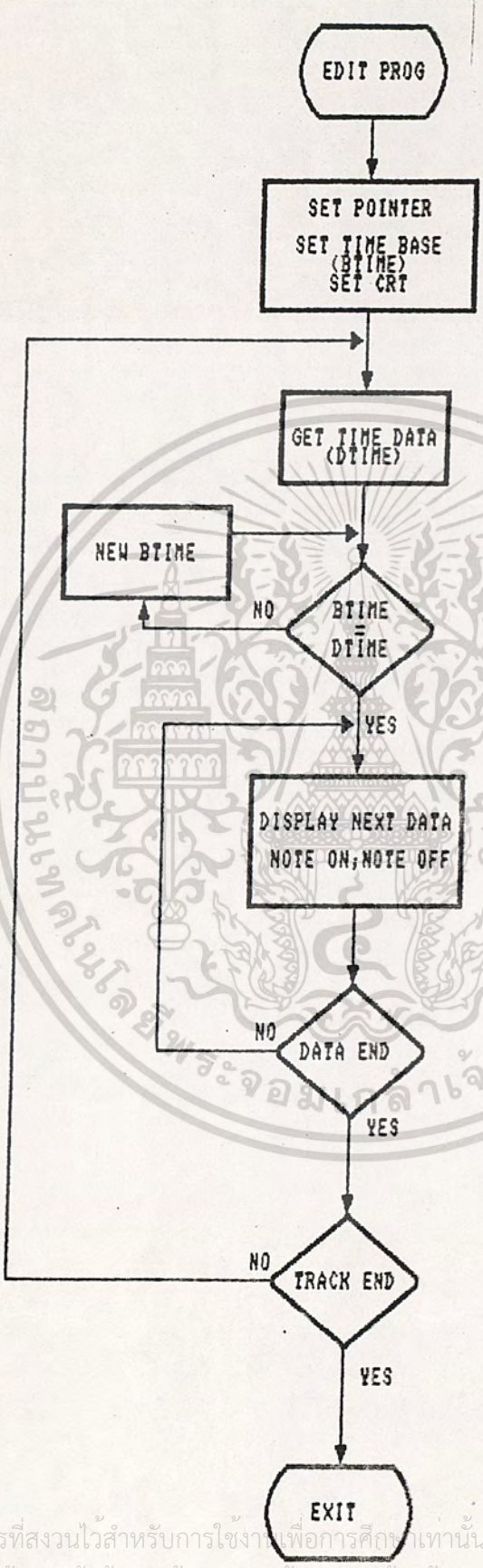
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟล์วชาร์ทแสดงส่วนบันทึกข้อมูล

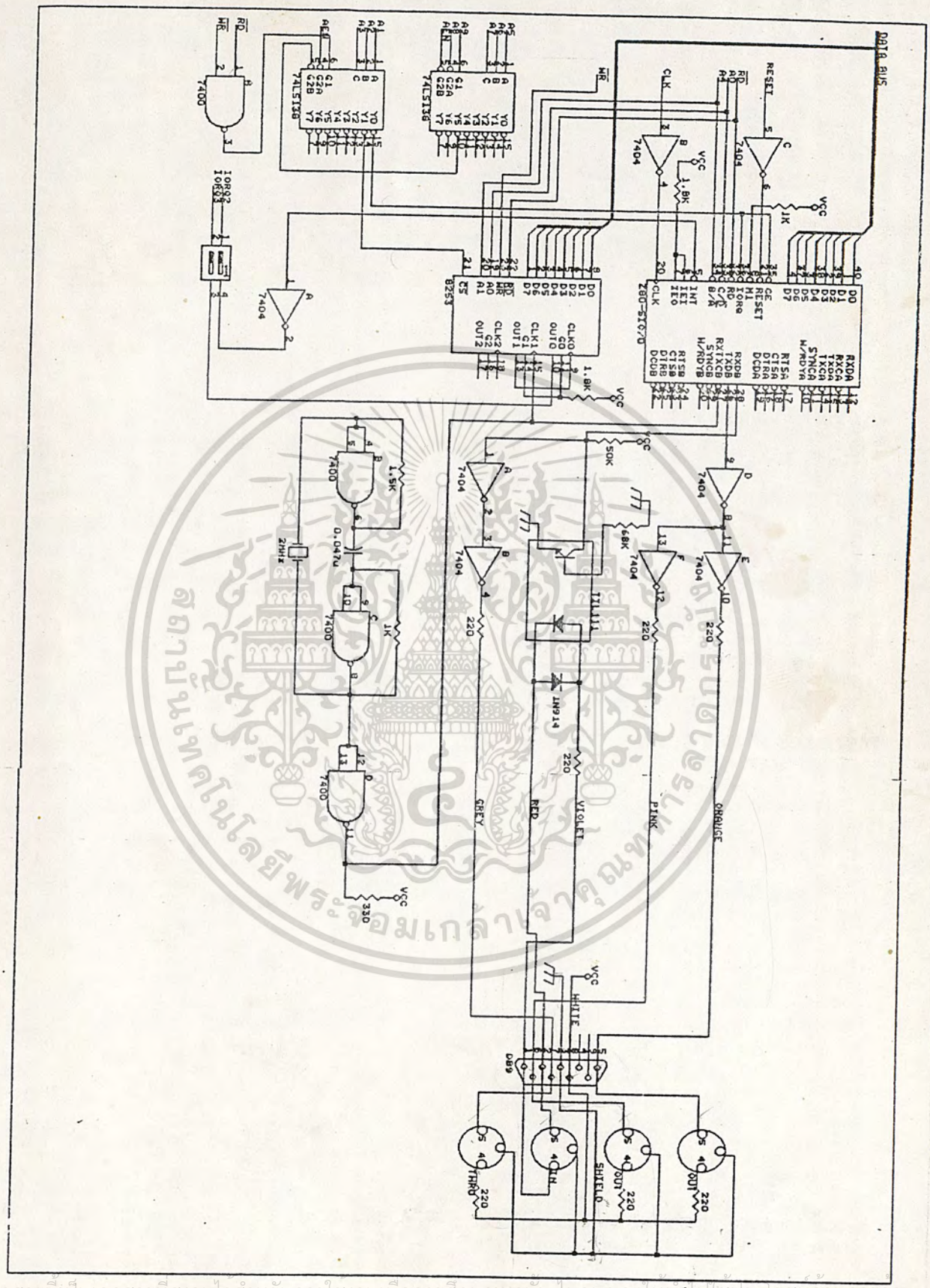


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟล์วชาร์ทแสดงส่วนแก้ไขข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 แผนภาพแสดงวงจรของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในขณะที่ทำการส่งข้อมูลทั้ง 16 แทรค (TRACK)แล้วก็จะทำการตีกลอง ซึ่งจะมีการทำงานร่วมกับโปรแกรม ดรัม แมชชีน ในการทำงานนี้

<u>ส่วน DRUM MACHINE</u>

จะทำการโปรแกรมจังหวะกลองต่าง ๆ โดยใช้รูปแบบ (PATTERN) ต่างๆ นำมาต่อเรียงกันเป็นเพลง และในขณะทำงานร่วมกับโปรแกรม ซีเควนเซอร์ (SEQUENCER) ก็จะทำหน้าที่ ตีกลองนี้ไปด้วย

<u>ส่วนปรับปรุงข้อมูล</u>

จะช่วยนักดนตรีแก้ไขปรับปรุง ข้อมูลเพลงในแต่ละ แทรค ให้มีประสิทธิภาพโดยจะมีฟังก์ชันดังนี้

- สามารถบันทึกข้อมูล ด้วยแป้นคีย์บอร์ด IBM ธรรมดา โดยการเขียนบนจอภาพในรูปแบบที่สะดวกในลักษณะ การแสดงผล เป็นรูป KEYBOARD และเลือกที่จะให้กด คีย์ใดที่เวลาใด
- สามารถบันทึกแบบ ซิงเกิล เสต็ป (SINGLE STEP) ซึ่งจะช่วยให้นักดนตรี สามารถบันทึกเพลงได้ทีละ เสต็ป เพื่อแก้ปัญหาเพลงบางเพลงที่เล่นได้ยาก
- มีระบบ ควอนไตซ์ ซึ่งจะช่วยจัดห้องเพลงและ รวบรัด เพลงให้ลงห้องให้ถูกต้องตามความจริงที่สุด
- สามารถ แทรก (INSERT),ลบ (DELETE) ท่อนเพลงแต่ละท่อนได้อย่างสะดวก

นอกจากโปรแกรมใหญ่ 3 ส่วนที่ได้กล่าวไปแล้ว ยังมีโปรแกรมการ โหลด(LOAD), เซฟ (SAVE) ข้อมูลลงบนแผ่นดิสเกตต์ (DISKETTE) โปรแกรมการ SETUP ค่าต่าง ๆ ของระบบในมิดี้

<u>หน่วยความจำ (MEMORY)</u>

โปรแกรมนี้จะแบ่งข้อมูลแต่ละ TRACK ไว้แต่ละ SEGMENT ใน MEMORY โดยวิธีนี้เราจะสามารถใช้บันทึกข้อมูลได้ TRACK ละ 64 KByte แต่ในการใช้งานจริงจะจอง MEMORY แต่ละ TRACK เพียง 10-20 KByte เท่านั้น

<u>FORMAT ของข้อมูล</u>

การบันทึกข้อมูลจะมีเวลา นำหน้าข้อมูลและตามด้วยข้อมูล มิดี้ จากนั้นตบท้ายด้วยรหัส (CODE) (FFH) ตัวอย่างเช่น

```
C300 90 60 7F 52 72 62 57 FF C700 80 56 72 FF ...
  |                        |  |          |
TIME 2 BYTE              END  TIME 2 BYTE  END
```

อย่างไรก็ดีการเซฟ ข้อมูลลง ดิสเกตต์ ก็ยังมีข้อมูลเกี่ยวกับการ SETUP ค่าต่าง ๆ จังหวะกลองลงแผ่นดิสเกตต์ ด้วย

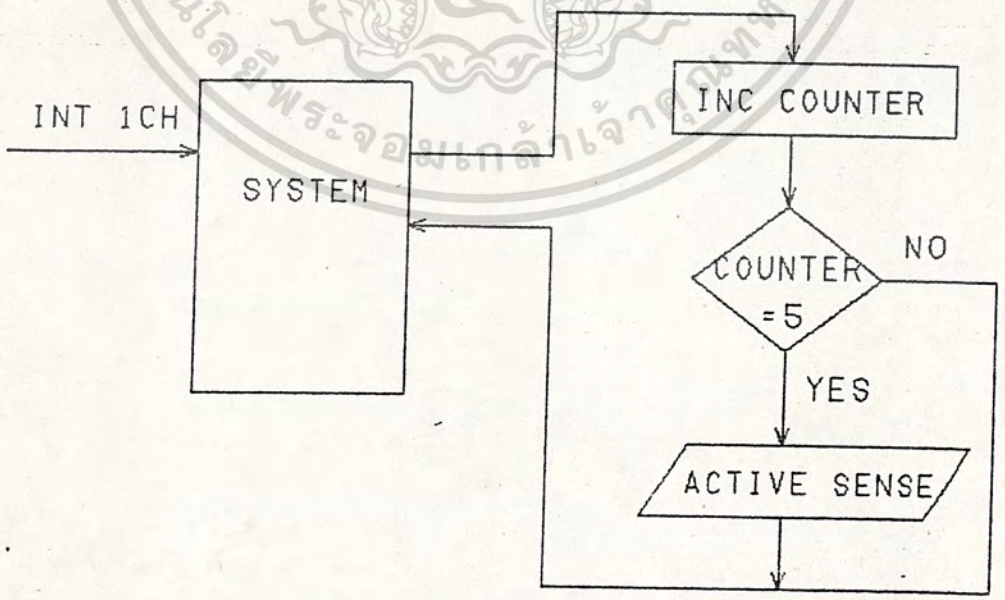
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟท์แวร์ ส่วนที่ทำงานร่วมกับ ฮาร์ดแวร์

ซอฟท์แวร์ ของโครงงานนี้จะทำงานประสานกับ ฮาร์ดแวร์ หลาย ๆ ส่วนในระบบซึ่งได้แก่

- คีย์บอร์ด ซึ่งใช้ในการติดต่อกับระบบทั้งในขณะเล่นเพลงและขณะ SETUP PARAMETER ต่าง ๆ ของเพลง รวมทั้งการ โหลด, เซฟ และ แก้ไข(EDIT)
- จอภาพ จะใช้การแสดงผล ข้อมูลต่าง ๆ ของระบบและที่สำคัญก็คือการ แสดงผลเวลา หรือหมายเลขบีท(BEAT) ของระบบ ซึ่งจะต้องใช้วิธี เติม (FILL) ลงใน หน่วยความจำหน้าจอ (CRT MEMORY)
- INT 1Ch ไทม์เมอร์ ติ๊ค อินเทอร์รัพท์ (TIMER TICK INTERRUPT) ตัวนี้จะทำงานขณะที่ 8253 ติ๊ค 8259เพื่อขอการเพิ่มค่าเวลาในระบบของ IBM/PC ซึ่งเราจะใช้โปรแกรมย่อย(RUTINE) นี้ในการส่ง แอ็คถีฟ เซนซ์ (ACTIVE SENSE) ลงไปใน อินเทอร์รัพท์ นี้
- MIDI CARD - 8253 INTERRUPT จะเป็นฐานเวลาสำหรับเพลงซึ่งจะถูก SET ให้มีขนาด 24 CLOCK/โน๊ตตัวดำ ผ่านทาง IORQ2
- MIDI CARD - SIO INTERRUPT จะ INTERRUPT ระบบเมื่อมีข้อมูลส่ง เข้ามาในระบบผ่านทาง IORQ3

ซอฟท์แวร์ ที่ทำงานเหล่านี้ จะทำงานร่วมกับ โปรแกรมหลัก(MAIN PROGRAM)ในขณะเล่นเพลง, บันทึกเพลง และการ SETUP ค่าต่าง ๆ สำหรับ คีย์บอร์ด และ มอนิเตอร์ เป็นการเขียนโปรแกรมติดต่อกับ ระบบผ่าน ไบออส (BIOS) และ DOS INTERRUPT ธรรมดา ซึ่งจะไม่ขออธิบายในที่นี้



รูปที่ 8 โฟลว์ชาร์ทแสดงโปรแกรมแอกทีฟเซนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INT 1CH

เนื่องจากความจำเป็นของ มิดี้ คอนโทรลเลอร์ (MIDI CONTROLLER)ที่จะต้องทำการส่ง แอ็คถีฟ เซนซ์ ให้กับอุปกรณ์ในระบบทุกตัว ในขณะที่ไม่มีการส่งข้อมูล มิดี้ ทุก 300 ms ดังนั้นเราจะใช้ INT 1CH นี้เนื่องจากมันจะอินเทอร์รัพท์ 18.2 ครั้งต่อวินาที ดังนั้นหมายความว่าถ้าเกิด INT 1CH ขึ้น 5 ครั้งติดต่อกันโดยไม่มีการส่งข้อมูลขึ้นเลย ก็ให้ส่ง แอ็คถีฟ เซนซ์ ออกไป

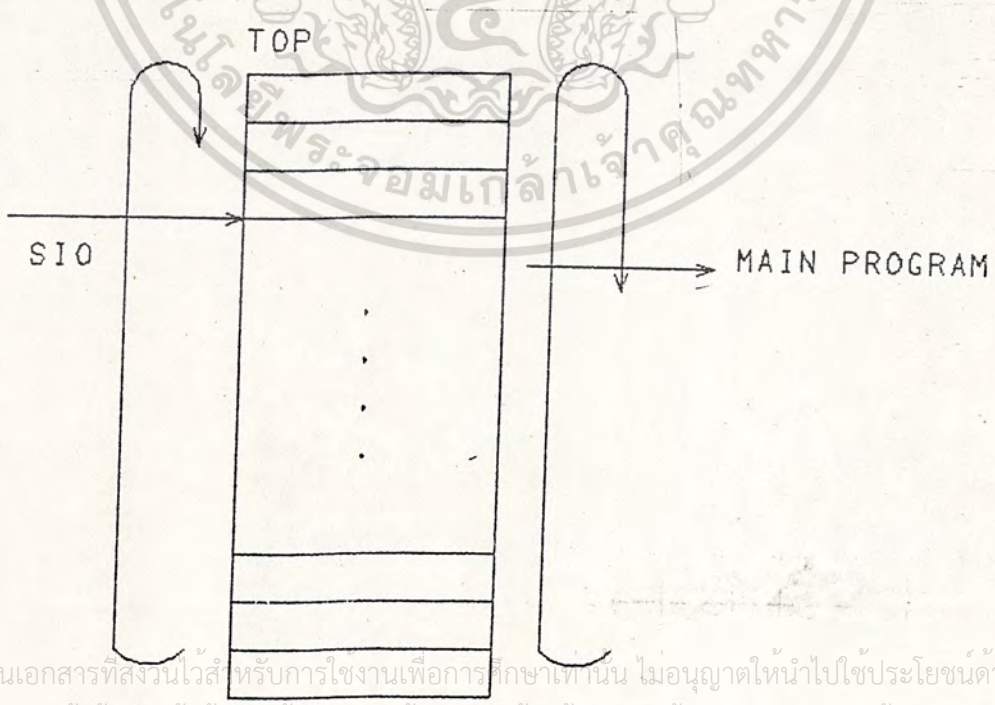
อย่างไรก็ดีขณะที่เราส่งข้อมูล มิดี้ ทั่วไปออกไปเราต้อง หยุด(BREAK)การนับอินเทอร์รัพท์ จำนวน 5 ครั้งนี้ใหม่ โดยการ SET วงจรนับ ที่นับให้เป็น 0 ใหม่ จากภาพขณะส่งข้อมูลก็จะทำให้ ค่าของตัวนับเป็น 0 เพื่อให้เริ่มนับใหม่หลังจากส่งข้อมูล มิดี้ ไปแล้ว

IORQ3 8253-MIDI CARD

ขณะที่เล่นเพลง เราจะต้องมีฐานเวลาที่แน่นอน ซึ่งเราสามารถกำหนดให้มีความเร็วตามมาตรฐาน มิดี้ คือ 24 CLOCK/1โน้ตตัวดำ ซึ่งอินเทอร์รัพท์ ตัวนี้จะทำการ เพิ่มเวลาของระบบแล้วส่ง รหัส ที่หมายถึง คล็อค ในที่นี้คือ 0F8H ออกไปด้วยและ IORQ3 จะทำการแสดงผล เวลาในหน่วยเป็น บีท ซึ่งในที่นี้ก็คือ 24 ครั้ง จะมีการแสดงโน้ต หนึ่งครั้ง

IORQ2 SIO-MIDI CARD

ขณะที่มีการบันทึก หรือมีการกดคีย์ ใด ๆ บนเครื่องคนตรีจะส่งรหัส มิดี้ ผ่านทาง SIO ซึ่ง SIO จะ INTERRUPT IBM/PC ผ่านทาง 8259 ที่ IORQ2 จากนั้นโปรแกรมก็จะอ่านข้อมูลเข้ามาบันทึกในแบบ หน่วยความจำแบบวงแหวน (CIRCULAR MEMORY) คือจะจอง หน่วยความจำไว้ 64 Byte แล้วเก็บวนไปเรื่อย ๆ โดยโปรแกรมใหม่จะทำหน้าที่อ่านแบบวงแหวน(CIRCULAR) ด้วย



รูปที่ 9 แผนภาพแสดงการทำงานของหน่วยความจำแบบวงแหวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างไรก็ดี ข้อมูลบางตัว เราไม่จำเป็นต้องใช้ เช่น แอ็คกีฟ เซนซ์, SYSTEM REAL TIME เพราะข้อมูลพวกนี้ตัว คอนโทรล ในระบบ มิดี้ จะเป็นตัวส่งเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลองและผลการทดลอง

ได้มีการทดลองนำเครื่องดนตรีประเภทมิดี้ หลายรุ่นหลายยี่ห้อมาทดลองใช้กับโครงงานนี้ และได้ทดลองผลิต มิดี้ คาร์ด ขึ้น 10 ชุด แล้วให้นักดนตรีและผู้ที่สนใจทดลองใช้ ปรากฏว่าได้ผลเป็นที่พอใจ ได้ทดลองใช้กับเครื่องดนตรีและคอมพิวเตอร์ต่าง ๆ กันดังนี้

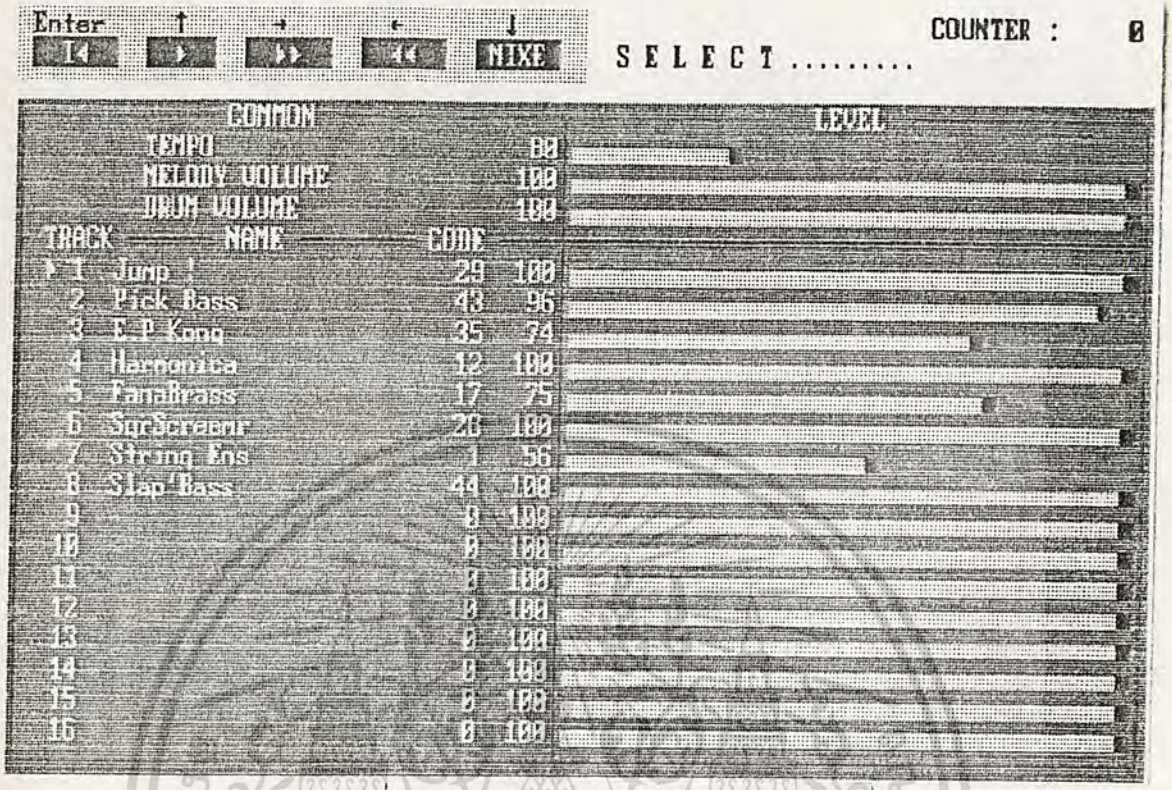
อุปกรณ์มิดี้	ไมโครคอมพิวเตอร์	อุปกรณ์แสดงผล
YAMAHA PSR-47 YAMAHA PSS-680 KAWAI K1 II KAWAI K4 ROLAND D5 YAMAHA TXZ-55	XT 4.77,10 Mhz AT 286 16,20 Mhz 386/SX/DX 28 Mhz	VGA EGA CGA MONOCHROME

ในส่วนของ ซีเควนเซอร์ สามารถบันทึกข้อมูลยาวถึง 22 นาที ในขณะที่ TEMPO เป็น 120 ซึ่งประสิทธิภาพของส่วน ซีเควนเซอร์ จะวัดได้จากการทำงานที่ TEMPO สูง แต่จากการทดลองส่งข้อมูลที่ TEMPO เป็น 300 ก็ยังไม่มีผลของการกระตุกของเพลงแต่อย่างไร และหน่วยความจำ ในแต่ละ TRACK ที่ใช้ก็ใช้เพียง 10 KBYTE เท่านั้น ในการบรรจุข้อมูลลงในดิสเกตต์ ก็ใช้เนื้อที่ไม่มาก เนื่องจากจะบันทึกข้อมูลจริง ๆ ใน 10 KBYTE นี้เท่านั้น

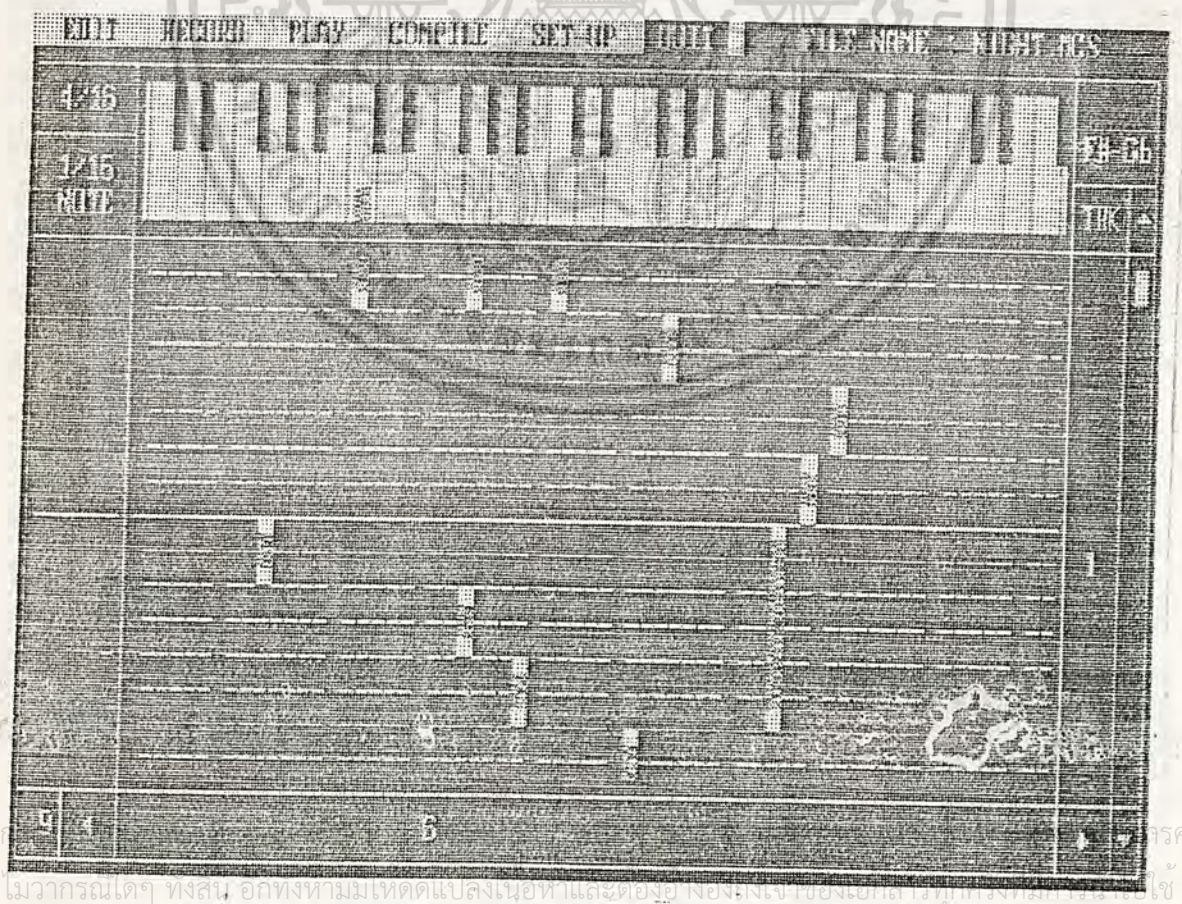
สำหรับส่วน ดรัม แมชชีน ได้ทดลองสอบถามจากผู้ใช้ถึงปัญหาในการใช้และประสิทธิภาพในการทำงานแล้วนำมาแก้ไขพบว่า ส่วน ดรัม แมชชีน สามารถสร้างจังหวะกลองได้อย่างไม่มีขอบเขต โดยในขณะนี้ยังไม่เคยพบว่ามีจังหวะกลองแบบใดที่ไม่สามารถทำการ EDIT ได้เลย แต่เนื่องจากทำงานอยู่ใน เทกซ์ โหมด (TEXT MODE) จึงไม่สามารถทำให้การพัฒนา บางส่วนกระทำได้ แต่ส่วน ดรัม แมชชีน ก็มีประสิทธิภาพเหนือกว่าโปรแกรมของต่างประเทศอยู่มากแล้ว

ผลการทดลองส่วนปรับปรุงและแก้ไขข้อมูล ได้ทำการแปลงข้อมูลทางมิดี้ มาอยู่ในรูปแบบช่วงเวลาการเล่นโน๊ต (SCALE TIME) เพื่อทำการ แก้ไข แบบเต็มจอ (FULL SCREEN) ผลการทดลองแปลงข้อมูล มิดี้ ทั้ง เพลง แต่ตัดมาให้ดูเป็นเสี้ยงบางห้องของเพลงเท่านั้น แต่ละแท่งที่เห็นจะบอก เวลาเริ่มต้นและจุดสิ้นสุดของเวลาการเล่นโน๊ตตัวนั้น ๆ ซึ่งสามารถนำมาเปรียบเทียบได้กับการใช้ โน๊ต ในลักษณะของ โน๊ตสากล (♩ ♪ ♫) ได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

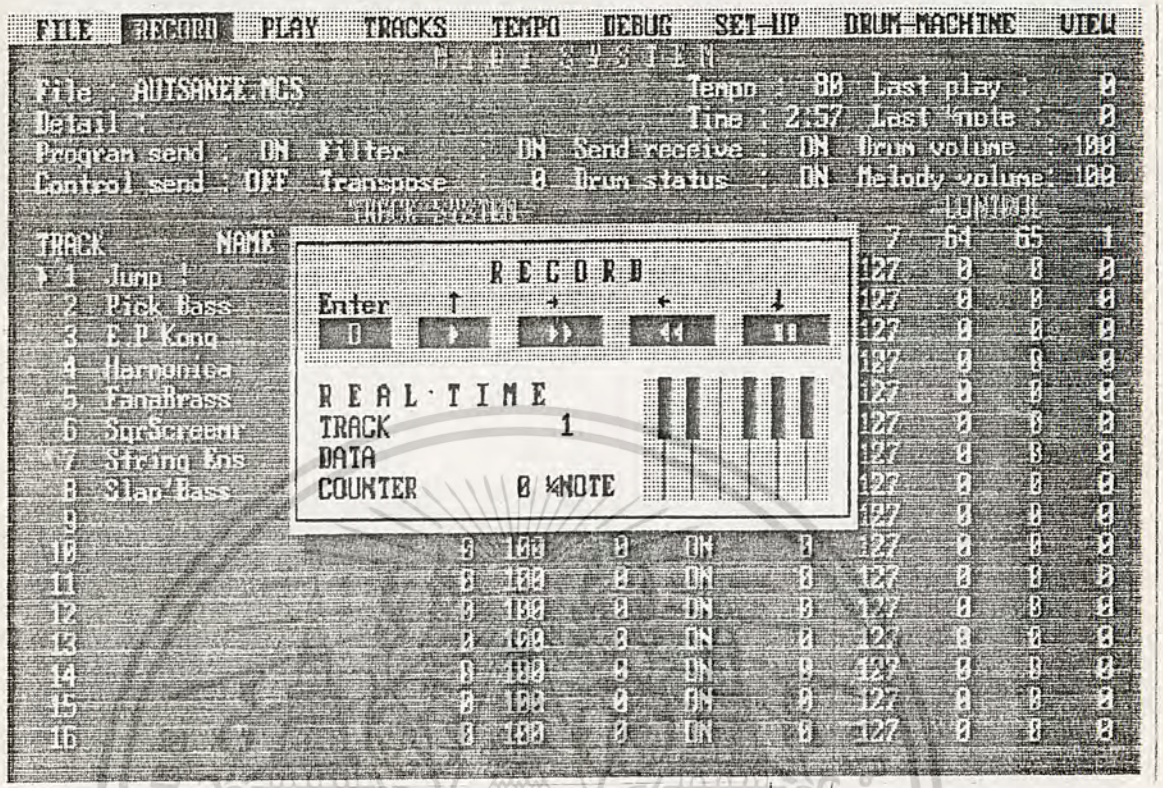


รูปที่ 10 ภาพแสดงโปรแกรมในขณะเล่นเพลง

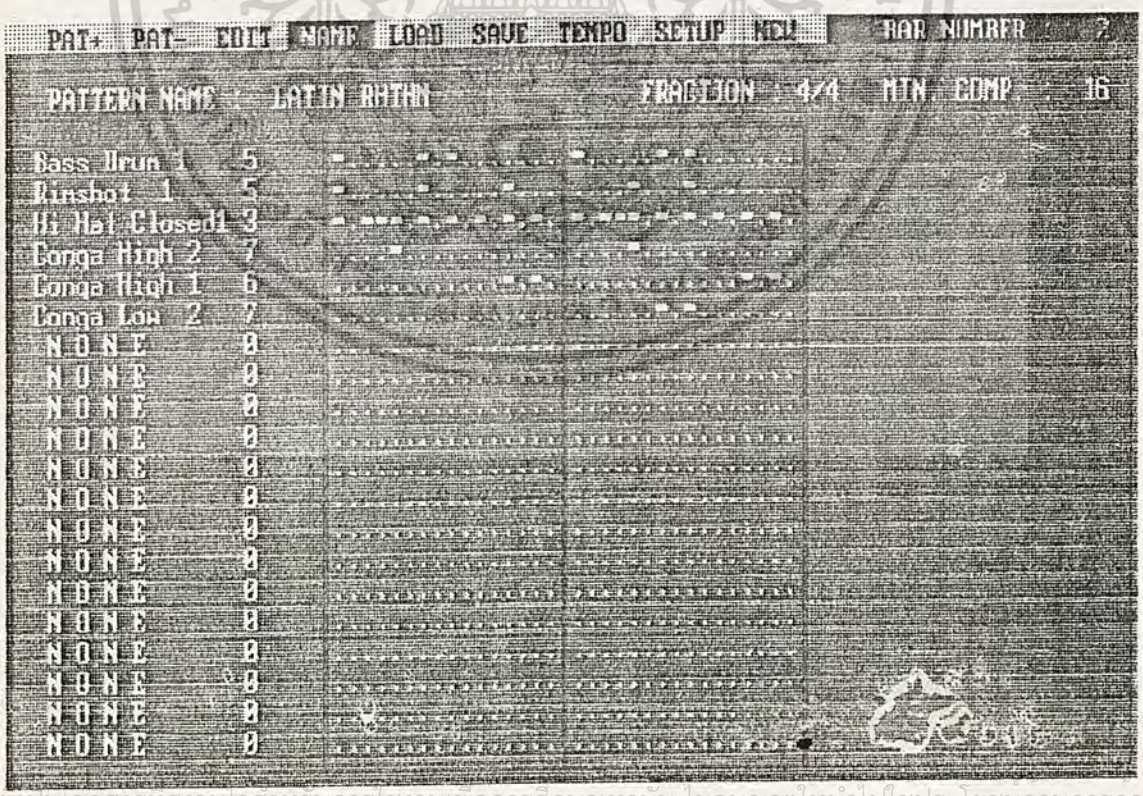


รูปที่ 11 ภาพแสดงโปรแกรมในขณะบันทึกข้อมูลผ่านคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปใช้เพื่อการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 ภาพแสดงโปรแกรมในขณะบันทึกข้อมูลทั่วไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 13 ภาพแสดงโปรแกรมในขณะบันทึกจังหวะรอง

บทที่ 5 สรุปและวิจารณ์

การนำ ไมโครคอมพิวเตอร์ ตระกูล IBM PC มาใช้สื่อสารในระบบ มิดี้ ก็นับว่าสามารถทำได้ดีเช่นเดียวกับไมโครคอมพิวเตอร์ ตระกูล APPLE, ATARI, COMMANDORE แต่ยังขาดประสิทธิภาพในด้านกราฟฟิคส์ (GRAPHICS) และความเป็นมาตรฐานของ เครื่องในแต่ละรุ่น เช่น ชนิดของจอที่ใช้แสดงผล แต่อย่างไรก็ดี ไมโครคอมพิวเตอร์ ตระกูล IBM PC ก็ได้เปรียบในแง่ความหลากหลายของ ซอฟท์แวร์ และราคา ดังนั้นการนำระบบ มิดี้ มาใช้กับเครื่องไมโครคอมพิวเตอร์ ตระกูลนี้ก็เป็นที่น่าสนใจมาก

โครงงานนี้สามารถนำไปใช้งานจริงได้แล้ว ซึ่งจะเป็นประโยชน์ต่อนักดนตรีทั่วไปเป็นอย่างมาก เนื่องจากประสิทธิภาพของตัวโปรแกรม และต้นทุนที่มีราคาถูกมาก เมื่อนำมาเทียบกับระบบที่มีขายอยู่ในท้องตลาด ซึ่งจะแยกจำหน่ายทั้งส่วน ฮาร์ดแวร์ และโปรแกรม ซึ่งมีราคาของ ฮาร์ดแวร์ สูงกว่าต้นทุนของโครงงานนี้ถึง 10 เท่า

แนวทางพัฒนาต่อ

1. พัฒนา ฮาร์ดแวร์ โดยใช้ ไมโครคอมพิวเตอร์ พวก ซิงเกิล ชิพ (SINGLE SHIP) แทน ซึ่งจะทำให้การทำงานเร็วขึ้น

2. พัฒนาส่วน ซอฟท์แวร์ ทั้งหมดให้เป็น โหมด กราฟฟิคส์ (GRAPHICS MODE) เพื่อประสิทธิภาพในการใช้งาน โดยเฉพาะอย่างยิ่งในส่วนแก้ไข กลอง และ โน๊ต แต่ต้องรักษาประสิทธิภาพการทำงานโดยคำนึงถึงเรื่องความเร็วเป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้คงไม่สามารถดำเนินลุล่วงไปได้ หากปราศจากการสนับสนุนจากบุคคลที่มีประคุณยิ่ง เหล่านี้

1. ผู้ก่อตั้งระบบ มิดี้ ซึ่งทำให้เกิดโครงงานนี้
2. ผู้ช่วยศาสตราจารย์ อาจารย์พลพุง ผดุงกุล อาจารย์ที่ปรึกษา
3. พี่เล็ก ผู้ให้คำแนะนำ และความช่วยเหลือต่าง ๆ ตลอดมา
4. รุ่นพี่ทุกท่านที่เคยทำโครงงานนี้มาก่อนได้ ช่วยวางแนวทางให้น้อง ๆ
5. บริษัท สยามกลการ จำกัด สำหรับคำแนะนำที่มอบคำแนะนำโครงงานนี้แบบไม่รู้ตัว
6. ชุมนุมคนตรี และวง 15 ออนซ์ ที่เป็นกำลังใจตลอดมา
7. ชุมนุมวิชาการ และห้องสมุดคณะวิศวกรรมศาสตร์ สำหรับความรู้ต่าง ๆ

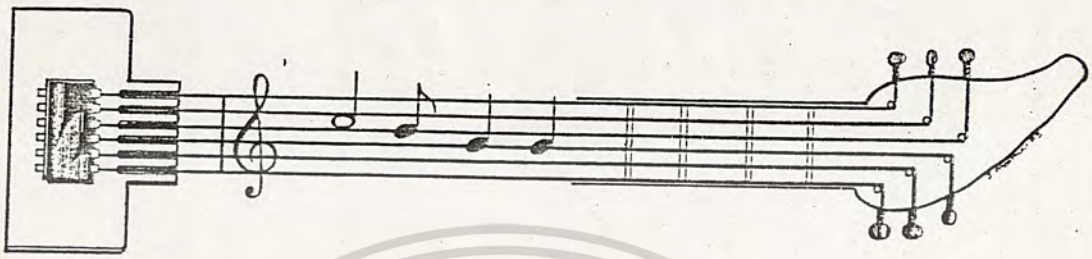
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIDI – Musical Instrument Digital Interface

Richard Evers, Editor



The idea started slow, but rapidly picked up speed.

Throughout the ages, man has always strived for new and exiting ways in which to amuse himself. Man is the master amuser. One way in which amusement was found, and still is today, is with music. In the beginning, there is a hint that the first musical instruments were rocks, producing a somewhat pleasing sound when smashed together in whatever unison they could muster. Soon, wood was found to possess the enthralling capacity to produce various sounds when struck. As time passed, man developed the art of cutting and forming pieces of wood to produce other truly unique sounds. From this simple beginning, we now have every instrument from the bongo drum to the grand piano. Over thousands of years of patient evolution, man has developed the most beautiful art of all – music.

This unique beginning is a primer to enable you to gain easy entrance into todays world of digital music synthesis, a world far removed from what music has been in past. Although the music synthesizer is capable of cloning the sounds of all man made instruments, it also has the ability to allow or disallow the limitations imposed on the instrument by mans capabilities. With a statement sure to be written off as sheer personal judgement, I will state that the finest of todays music synthesizers are capable of mimicking the sounds of man made instruments to such an extent that the synthetic counterpart will be preferred. But, as with all new concepts, many people will strive to retain the older, more familiar yet less refined methods of past.

Today, in what is known as the age of the microprocessor, the true "State Of The Art" (what does that mean?) phase of our existence, analog systems still persist in great numbers. Although the digital movement has taken all forms of music reproduction by storm, ie. the compact disk, digital recorded record albums and, of course, music sythesis, analog is still with us. Time will be the sole test of our loyalty to a friend long past its prime. Achieving results that were once thought impossible using analog logic now appear common place due the advent of todays digital high technology. But, justly enough, within a few years travel, our current high tech will also be considered obsolete. The sad fate of time. Obsolescence.

To get on track, a few years ago, when digital synthesis was beginning to reach its apex, a very obvious road block materialized. There were no standards to meet, no industry standard that would allow various components to be easily connected together for use. The manufacturers were producing terrific products, but the buyer was restricted to one name brand for add ons. A restricting situation.

To solve this problem, a few key people in the music sythesis industry started thinking, talking, and attracting attention along this line. The idea started slow, but rapidly picked up speed. MIDI, The Musical Instrument Digital Interface, was the solution presented to combat this problem.

The MIDI system is a series of guidelines that manufacturers should meet in order to ensure the compatibility of their products with others. As with all set guidelines, MIDI is not perfect and does actually impose limitations at times. Without coming down too hard at first, let me explain the guidelines.

The system is based on the almost current 8 bit technology, mainly because the designers were trying to produce a system that would easily fit into the budget of most starving artists. They chose a universal cable connection of the 5 pin DIN plug, using only 3 of the 5 pins. Pin #2 is ground, with pin #'s 4 and 5 being used for a current loop.

" The interface operates at 31.25 kilobaud (±1%), asynchronous, with a start bit, 8 data bits (D0 to D7), and a stop bit. This makes a total of 10 bits for a period of 320 microseconds per serial byte. "

The above paragraph was lifted directly from the MIDI 1.0 Specifications, Document No. MIDI-1.0, Dated August 5, 1983.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

With each MIDI equipped unit, you will notice either two or three connectors for MIDI use. There is the MIDI In, MIDI Out, and possibly, MIDI Through (American spelling Thru). MIDI In is the connector for data coming into the unit from other MIDI equipped units on line. This can include drum and rhythm machines, extra keyboards, electric guitar and drum interfaces, controllers and sequencers, or extra synthesizers.

A total of 16 channels, numbered 1 through 16, are allowed under MIDI but more than 16 instruments can be on line at any one time. By assigning duplicate channel numbers to units, or having some units respond to all messages (see Omni Mode), vast numbers of units can be serviced. One main limitation to this provision is the extended line length required, which produces added line loss and a greater time delay with transmitted and received data. By reading more than a few articles on this subject, one fact becomes apparent. When using more than a couple of MIDI units on line, noticeable delays appear in what is deemed simultaneous sound reproduction. But, I have recently read an outstanding interview with a person well qualified in the field. He stated that the time delay between units was not completely the fault of the MIDI implementation. In his opinion, the rate at which the units process the information provided leads to the noticeable time lag. Without greater experience in the subject, it is hard to take a side.

MIDI Out is the data being transmitted from the MIDI equipped unit. MIDI Through is a copy of the information currently being passed into the unit through MIDI In. This feature, if provided, gives the user the ability to slave units together. According to MIDI specs, the cable length between units cannot exceed 50 feet, using shielded, twisted cable with Pin #2 connected to the shielding at both ends.

The transmitted data is quite easy to identify. All transmitted information is 8 bits in length, with the Status bytes having the high bit set and data bytes having the high bit clear. As we will soon discuss, the Status byte has been provided to control the system. Within the Status class will be found all commands necessary to determine what the system will do, when it will do it, and how it should be done. The Data bytes are really the workers of the system. Not only do they carry information such as the note value to be played, but they also labour under the rule of the Status bytes to ensure that all commands are understood. For the purpose of this article, the eight bit byte is arranged from bits 7 to 0, with bit 7 acting as the high bit.

The charts below have been prepared to show all the possible byte values encountered while working with the MIDI system. Complete explanations will follow.

Table Of Notes Corresponding To Data Values
Expressed In Hex

	Notes											
Octave	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
–	$00	$01	$02	$03	$04	$05	$06	$07	$08	$09	$0A	$0B
0	$0C	$0D	$0E	$0F	$10	$11	$12	$13	$14	$15	$16	$17
1	$18	$19	$1A	$1B	$1C	$1D	$1E	$1F	$20	$21	$22	$23
2	$24	$25	$26	$27	$28	$29	$2A	$2B	$2C	$2D	$2E	$2F
3	$30	$31	$32	$33	$34	$35	$36	$37	$38	$39	$3A	$3B
4	$3C	$3D	$3E	$3F	$40	$41	$42	$43	$44	$45	$46	$47
5	$48	$49	$4A	$4B	$4C	$4D	$4E	$4F	$50	$51	$52	$53
6	$54	$55	$56	$57	$58	$59	$5A	$5B	$5C	$5D	$5E	$5F
7	$60	$61	$62	$63	$64	$65	$66	$67	$68	$69	$6A	$6B
8	$6C	$6D	$6E	$6F	$70	$71	$72	$73	$74	$75	$76	$77
9	$78	$79	$7A	$7B	$7C	$7D	$7E	$7F				

Status Bytes With Messages

Bit Pattern	Decimal Value	Hex Value	Data Bytes	Status Byte Message
1000 xxxx	128 to 143	$80 to $8F	2	Note Off
1001 xxxx	144 to 159	$90 to $9F	2	Note On
1010 xxxx	160 to 175	$A0 to $AF	2	Polyphonic Key Pressure After Touch
1011 xxxx	176 to 191	$B0 to $BF	2	Control Change
1011 xxxx	176 to 191	$B0 to $BF	2	Select Channel Mode
1100 xxxx	192 to 207	$C0 to $CF	1	Program Change
1101 xxxx	208 to 223	$D0 to $DF	1	Channel Pressure After Touch
1110 xxxx	224 to 239	$E0 to $EF	2	Pitch Wheel Change
1111 0000	240	$F0	variable	System Exclusive
1111 0001	241	$F1	–	Unimplemented
1111 0010	242	$F2	2	Song Position Pointer
1111 0011	243	$F3	1	Song Select
1111 0100	244	$F4	–	Unimplemented
1111 0101	245	$F5	–	Unimplemented
1111 0110	246	$F6	0	Tune Request
1111 0111	247	$F7	0	End Of Exclusive
1111 1000	248	$F8	0	Timing Clock
1111 1001	249	$F9	–	Unimplemented
1111 1010	250	$FA	0	Start
1111 1011	251	$FB	0	Continue
1111 1100	252	$FC	0	Stop
1111 1101	253	$FD	–	Unimplemented
1111 1110	254	$FE	0	Active Sensing
1111 1111	255	$FF	0	System Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

After that unrelenting barrage of MIDI information, I feel that it is only fitting to supply you with a bit more of a detailed description. To begin, the frequency data bytes, hex values $00 to $7F, are obviously the note values as expressed through almost 10 complete octaves. This is one drawback of the system. 7 bits of resolution is not sufficient to allow graduations in smaller increments than semitones. One recent speculation made regarding this point stated that future revisions of the system specifications should allow note increments in graduations of cents, 1/100's of a semitone. A possible but difficult to implement idea.

Status Messages, in contrast to the note values, require more than a little explanation before becoming coherent to any degree. Therefore, there are two different Status messages that can be encountered. The Channel Message and the System Message.

The Channel Message

A Channel message is one that states a message and a specific unit number to which the message is addressed. The lower 4 bits, bits 0–3, are used for this identification. In the charts shown above, plus the descriptions to follow, this has been signified by the xxxx in place of the lower nibble in each appropriate bit pattern. This 4 bit channel designation also explains the limitation of this system to 16 channels (decimal range 0–15).

There are two types of channel messages; Voice and Mode. Voice messages control each separate units voices, which are sent over the voice channels. The Mode message defines the instruments response to the Voice messages, sent over the instruments basic channel. The Mode message can control such things as Local Control On and Off, All Notes Off, Omni Mode On or Off, Mono Mode On/Poly Mode Off, or Poly Mode On/Mono Mode Off. See message Select Channel Mode for more information.

The System Message

System messages are ones that do not pertain to any one specific channel. Being the elite of the system entities them to speak and except all to listen, immediately. Within the realm of the System message falls the Common, Real-Time, and Exclusive messages.

Common messages are directed to all units, regardless of channel number. The Common messages are comprised of Song Position Pointer, Song Select, Tune Request, and End Of Exclusive. Each Common message consists of a single byte, of which an in depth explanation will follow promptly.

Real-Time messages are also intended to be heard by all units in the system, at any time, even during the transmission of other data by a Status byte. A Real-Time message is one in which can be tested for and ignored by the units or acted upon if the information is desired. Real-Time messages are comprised of the Timing Clock, Start, Continue, Stop, Active Sensing, plus System Reset. Explanations will follow.

There is only one Exclusive Message, of which carries one Status byte, plus any length of data bytes following. It is terminated by either the Common message End Of Exclusive (EOX), or any other Status byte. An Exclusive Message is one that has been incorporated to identify the manufacturer of each piece of equipment, plus allow the manufacturer to transmit whatever message they please. A form of personalized service, finally.

Status Byte Explanations

Message:**Note Off**
Pattern: 1000 xxxx

The Note Off Message, used in conjunction with the Note On message, determine when each note will start and stop for the channel affected. Following this byte are always two data bytes. The first byte determines which note within a specific octave is to be affected, as can be demonstrated in the note chart above. The second byte is the Note Off Velocity. This value is used in synthesizers capable of ADSR (attack, decay, sustain, release), to set the rate of release from the sustained level. In synthesizers not so enabled, the release is immediate.

Message:**Note On**
Pattern: 1001 xxxx

As specified above, the Note On command is used to determine which note is stuck on a specified channel. The Note On Velocity, the second data byte, has a unique use this time. It determines the loudness setting of the note value struck. In units capable of ADSR, this level of loudness will be attained within a user predetermined time limit. If the unit is incapable of ADSR, the level will be reached immediately. The maximum loudness setting, considering we are working with the lower 7 of 8 bits, is a decimal value of 127. A value of 0 would be the same as turning the Note Off. According to specs, a value of 1 is equivalent to "triple pianissimo", very quite. A value of 127 (maximum), is "triple forte", very loud. The middle of the scale is 64, which is somewhere between "mezzo–piano" and "mezzo–forte" – middle scale. If the unit is incapable of various velocity settings, a value of 64 is used and transmitted.

Message:**Polyphonic Key Pressure After Touch**
Pattern: 1010 xxxx

Polyphonic Key Pressure After Touch is really just what is implied. The new value of key pressure attained after touching it. If the value has been set before to reflect the key pressure exerted, and the user strikes the key once again, this new pressure must be reflected. This status byte has two data bytes following. The first byte is the note stuck, with the second value byte representing the pressure. The values of pressure range from 0, no pressure at all, to 127, bashed through the keyboard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message:**Control Change**
Pattern: 1011 xxxx

Control Change is a status byte that is transmitted whenever a controller mechanism is adjusted. Controller mechanisms in this context refer to foot pedals, knobs (pots), modulation wheels, sliders, and switches. Two data bytes are used after the status message to reflect this change. The first byte indicates which controller was affected, while the second byte determines the new value attained. There are four ranges allowed for various controller mechanism used, as shown below.

Value Range	Controller Type
$00 to $3F	Continuous (ie. Foot Pedals, Knobs, Modulation Wheels, Sliders)
$40 to $5F	Switches
$60 to $7A	Presently Undefined
$7B to $7F	Channel Mode Messages (see Select Channel Mode following)

The first range, Continuous, often requires a more subtle range of graduations in comparison to say a switch. Either a switch is off or on, 0 or greater than 0. Continuous devices such as a slider can graduate across the scale in as large or small increments as desired, therefore a very subtle range is required. For this reason, the Control Change message allows for three data bytes if necessary. As stated before, the first data byte determines the controller affected, with the second data byte determining the value to set. This acts as the High Byte, or increment of 256 Lower Bytes. By now you know that the third byte is the Low Byte, thereby allowing a range of 14 bits, or 16,384 graduations. The Low Byte is not mandatory, therefore it can be left off if not required.

Message:**Select Channel Mode**
Pattern: 1011 xxxx

Select Channel Mode is a unique message that allows alterations to the way a MIDI unit will respond to, and transmit MIDI channel messages. Two data bytes are used to allow this transformation, with the first data byte having a limited range of $7A to $7F. This limitation is imposed due to the fact that two different messages share the same bit pattern (see Control Change above). The second data byte is used in conjunction with the first to achieve the result desired, as shown below.

Byte 1	Byte 2	Result
$7A	$00	Local Control Off
$7A	$7F	Local Control On
$7B	$00	All Notes Off
$7C	$00	Omni Mode Off, All Notes Off
$7D	$00	Omni Mode On, All Notes Off
$7E	$0x	Mono Mode On, Poly Mode Off, All Notes Off (x = number channels)
$7F	$00	Poly Mode On, Mono Mode Off, All Notes Off

The term Poly refers to Polyphonic sound reproduction, which is the ability for the unit to allow more than one note to be played simultaneously. Mono refers to Monophonic sound reproduction, which is reproduction of sound that only allows one note to be played during any one time period.

Omni is a term that is used to describe the ability for a unit to respond to all system messages, or only ones addressed to it's basic channel. When Omni is Off, the unit will only listen for messages addressed to itself. When Omni is On, it will listen and act upon every message coming over the bus.

Local Control is the ability for a unit to act through, or bypass it's own circuitry for the generation of sound. With Local Control On, the synthesizer will act like it normally does. With Local Control Off, the keyboard will still produce MIDI data as it is played, but the synthesizer will not produce any sound to compliment the data. This feature was incorporated to allow keyboards to control instruments other than the synthesizer attached.

Message:**Program Change**
Pattern: 1100 xxxx

Program Change is used with synthesizers that have banks of memory set aside for various user chosen sounds. Often times these sounds are assigned by changing the bank or patch number currently in use. Program Change allows a reflection of a change in this bank number for the channel affected, with one data byte assigned for the bank number chosen. The bank number can have a value of 0 to 127.

Message:**Channel Pressure After Touch**
Pattern: 1101 xxxx

The Channel Pressure After Touch is an interesting feature that determines the average pressure values for the unit at any instant. With this average pressure determined, a desired variation of the overall timbre value or volume of the instrument can be calculated by determining the deviation from the average required. The single data byte allowed can have a value of 0, no pressure, to 127, as much pressure as possible all around.

Message:**Pitch Wheel Change**
Pattern: 1110 xxxx

This message is one that allows a reflection of any change in the setting of the pitch wheel. As with Continuous Controller Mechanisms such as sliders, a fairly large scale is required to reflect variations in the pitch wheel setting. For this purpose, 14 bits of resolution have been provided. One odd point to note about the MIDI system specs at this point. With the Control Change message, a resolution of 14 bits was available if required. The data bytes were read High to Low. For a Pitch Wheel Change, 14 bits are also provided, but they are read Low to High. This poorly thought out variation could lead to confusion in writing code if taken for granted. So much for conventions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message:**System Exclusive**
Pattern: 1111 0000

A System Exclusive message is one that I find to be refreshing. It allows identification of the manufacturer, and also allows the same to get his two bits in. A System Exclusive message is comprised of the first data byte signifying the identification number assigned to the manufacturer, as per the IMA (International MIDI Association In California), and as many bytes following as the manufacturer requires to tell his story. This message is terminated either by an End Of Exclusive Message (EOX – 1111 0111), or any other Status byte that happens by. Below can be found a partial list of manufacturers ID codes, as supplied by the IMA.

Manufacturer	ID
Sequential Circuits Inc.	$01
Big Briar	$02
Octavew/Plateau	$03
Moog Music	$04
Passport Designs	$05
Lexicon	$06
Bon Tempi	$20
S.I.E.L.	$21
Kawai	$40
Roland	$41
Korg	$42
Yamaha	$43

Message:**Song Position Pointer**
Pattern: 1111 0010

The Song Position Pointer is a 14 bit value that allows a record to be kept of the number of beats since the start of a song session. This has been incorporated for use with synthesizers equipped with a sequencer (digital recorder), or for a rhythm machine. With this feature enabled, a flag can be set to allow music to be played from a specific location within the song. The two data bytes are read Low to High.

Message:**Song Select**
Pattern: 1111 0011

As with the Song Position Pointer, Song Select is also meant to be used with synthesizers equipped with a sequencer, or for a rhythm machine. Song Select uses one data byte to select which song or note sequence is to be played after a Start message has been received.

Message:**Tune Request**
Pattern: 1111 0110

Tune Request is a throw back into the age of Analog synthesizers. This single Status byte, no data bytes, is used to request a tuning of the Analog synthesizers oscillators.

Message:**End Of System Exclusive (EOX)**
Pattern: 1111 0111

The EOX message is a single Status byte, no data bytes, that flags when a System Exclusive Message has been completed.

Message:**Timing Clock**
Pattern: 1111 1000

The Timing Clock Message is one that can be used to synchronize all sequencers and/or rhythm machines on line. The clock transmits its message at a rate of 6 messages per beat. As stated earlier, this type of message will appear at a regular intervals, regardless of the current state of other Status messages.

Message:**Start**
Pattern: 1111 1010

As before, this Status message is intended for use with a synthesizer equipped with a sequencer or rhythm machine. This message will inform the sequencer/rhythm machine to begin playing a pre-arranged song or note sequence from the beginning. There is only one Status Byte, no data bytes, for this message. See the Song Select Message for a little more information regarding Start.

Message:**Continue**
Pattern: 1111 1011

As before, the Continue message has been incorporated for use with synthesizers equipped with a sequencer or rhythm machine. This message is used to restart the current song sequence after receiving the next Timing Clock message. The sequence is then picked up from the next position in the Song Position Pointer. This message is flagged by the user pressing the Continue button on the sythesizer. As before, this message carries one Status byte and no data bytes.

Message:**Stop**
Pattern: 1111 1100

Again, the Stop message is for synthesizers equipped with a sequencer or rhythm machine. When received, this message tells the sequencer to stop playing its current sequence. This message carries one Status byte and no data bytes.

Message:**Active Sensing**
Pattern: 1111 1110

This message is one that is transmitted by any MIDI instrument on line, powered up, but not actively involved in anything. This message is transmitted once every 300 milliseconds if

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

there is no activity on the MIDI bus. One Status byte and no data bytes.

Message:**System Reset**
Pattern: 1111 1111

The System Reset message performs exactly as you might expect. It tells all MIDI instruments on line to perform a power up sequence to return them to a freshly powered up state.

Retrospect

The MIDI system, as has been expressed throughout this article, is a series of well thought of specifications. Through each manufacturers implementation, musicians can use instruments of different origin and expect predictable results. This is the basic flaw in the industry today. MIDI on paper appears quite explicit in its goals. But soon after implementation, the manufacturers discovered many minute points not completely taken into consideration by MIDI Version 1.0. Today, it is this problem that confronts every musician who considers the step into the world of music synthesis.

In partial explanation, manufacturers immediately found problems implementing MIDI 1.0 following its release. Although each manufacturer tried their best to work the MIDI system within the boundaries of their machines, hindsight informed us of the inevitable. The manufacturers did not work together to make sure the systems were compatible. Each operated within their own collective vacuum, producing equipment meeting untested specifications, without considering the fact that MIDI 1.0 might be vague enough to allow multiple interpretations.

When the first MIDI machines were released on the market, problems became apparent immediately. Although many of the machines were compatible, some subtle to extreme cases of incompatibility did exist. And, as with any manufacturer faced with high R&D costs, plus further misinterpretations, many of the problems went unresolved. Take for example the Yamaha DX7 music synthesizer. It's a great machine, yet its MIDI implementation is not completely compatible with say a Roland or a Korg. In this relatively new field of digital music synthesis, it is not hard to find people who have personally become victims in this rush to compatibility.

Although I may appear to hold conflicting opinions regarding the MIDI, this is not so. MIDI in general does have its fine points. But I think that it is time for the manufacturers to get together and try to work their problems out. The specifications do not have to be altered just yet, just clarified down to the finest detail. Following this, each manufacturer can regroup to produce low cost true MIDI updates for the machines currently out. The longer this move is neglected, the greater the chance of more permanent problems. So much for a bit of sage advice.

In Summation

In closing, I would like to reflect on possible extensions to the MIDI 1.0 specifications. Due to the fact that 16 and 32 bit chips have dropped in price, implementations using these chips could be considered. With these chips, high speed communications, multi tasking, and unbelievable control over vast amounts of RAM and ROM could be taken advantage of. Further to this, we have had a chance to work with the MIDI 1.0 for quite a while now. This time of reflection has enabled musicians from all over to discover most of the weak points inherent in the system. With these points in mind, plus the technology available to us today, a true implementation could be performed, with one problem. The market has already been flooded with MIDI 1.0 equipped units. A revision at this stage would make everything else obsolete, as far as current thought allows. Due to this single fact, the MIDI specifications will not be allowed revision on a radical scale for some time to come. Although it may be hard to accept this twist of fate, the human side of this story requires consideration. Musical technology is slated to stagnate for the next few years, after which time will come an age of radical change. What our dreams are composed of today is the reality of tomorrow.

Finally, I would like to thank a few people who have helped me understand MIDI more than I ever thought possible. They include Vera Barycky, who supplied me with vast amounts of hard to get information that otherwise would be inaccessible. My father, Ted Evers, for his continuous stream of information and knowledge, and my brother, John Evers, for all his related experience working within the field, from which I was able to extract some particularly critical information.

References

The MIDI Handbook	by Paul Vytas
MIDI Specification 1.0	by the IMA in California
The Roland Juno–60 Manual	by Roland
The Roland JX–3P Implementation	by Roland
The Yamaha RX15 Digital Rhythm Programmer Manual	by Yamaha
Keyboard Magazine	June 84 & July 85
Mix Magazine	August 1984
Computing Now	December 1984
A stack of IMA newsletters	by the IMA in California

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Synthesizer
MODEL K4/K4r

MIDI Implementation Chart

Date: Aug. 1989
Version: 1.0

Function ...		Transmitted	Recognized	Remarks
Basic Channel	Default Changed	1—16 1—16	1—16 1—16	Memorized
Mode	Default Messages Altered	— X * * *	1, 3 OMNI ON/OFF	Memorized MONO ignored
Note Number	: True Voice	24—108 * * *	0—127 0—127	
Velocity	Note ON Note OFF	* *	* *	
After Touch	Key's Ch's	X *	X *	
Pitch Bender		*	*	
Control Change	1 7 64 100, 101 6	* X * * (0, 1) *	* * * * (0, 1) *	Modulation Volume Hold 1 RPC Data entry
Prog Change	: True #	* * * *	* 0—127	
System Exclusive		*	*	
System Common	: Song Pos : Song Sel : Tune	X X X	X X X	
System Real Time	: Clock : Commands	X X	X X	
Aux Messages	: Local ON/OFF : All Notes OFF : Active Sense : Reset	X O (123) O X	O O (123-127) O X	
Notes		* Can be set to O or X Memorized even after turning off the power RPC #0=Pitch Bender sensitivity #1=Master fine tuning Values are given by Data entry		

Mode 1 : OMNI ON, POLY
Mode 2 : OMNI ON, MONO
Mode 3 : OMNI OFF, POLY
Mode 4 : OMNI OFF, MONO

O : Y
X : N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[PORTABLE KEYBOARD]

Model PSS-680 Mode 99 MIDI Implementation Chart Version: 1.0

Function		Transmitted	Recognized	Remarks
Basic Channel	Default Changed	1-16 (*1) 1-16	1-16 (*2) 1-16	
Mode	Default Messages Altered	3 × **********	3 (*9) × ×	
Note Number:	True voice	24-108(*3) **********	24-108 24-108 (*4)	
Velocity	Note ON Note OFF	× 9nH, v = 1-127 (*5) × 9nH, v = 0	○ 9nH, v = 1-127 × 9nH, v = 0 /8nH	
After Touch	Key's Ch's	× ×	× ×	
Pitch Bender		○	○	7bit Resolution
Control Change	1 64 65	○ ○ ○	○ ○ ×	VIBRATO SW SUSTAIN SW PORTAMENT SW
Prog Change :	True	○ (*10) **********	○ 0-127 (*11) 0-99	
System Exclusive		○	○	
System Common	: Song Pos : Song Sel : Tune	× × ×	× × ×	
System Real Time	: Clock : Commands	× ×	× ×	
Aux Mes-sages	: Local ON/OFF : All Notes OFF : Active Sense : Reset	× × ○ ×	× × ○ ×	

Note: (*8) For the channels from 1 to 15, 0-99, 100-104 and 105-127 are recognized corresponding to the voices #00-99, the DIGITAL SYNTH BANK 1-5 and voices #00-22 respectively. For the 16th, 0-99 and 100-127 correspond to Style #00-99 and Style #00-27 respectively.

(*9) Multi timbre, Dynamic allocation.

(*10) For transmission, 00-99 and 100-104 correspond to voices #00-99 and the DIGITAL SYNTH BANK 1-5 respectively in the VOICE SELECTION. In the STYLE SELECTION, 00-99 are transmitted via Ch16 corresponding to #00-99.

(*11) Channels 1 through 15 recognize 0-99, 100-104 and 105-127 as voices #00-99, the DIGITAL SYNTH BANK 1-5 and voices #00-22. Ch16 does not recognize any signals.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. CRAIG ANDERTON, "MIDI FOR MUSICIANS." AMSCO PUBLICATION 105P.1986
2. JAY KUBICKY "A MIDI PROJECT", 199P. BYTE JUNE 1986 VOL 11
3. IBM PC/XT TECHINICAL REFERENCE
4. MARCRO ASSEMBLER V.4.0
5. คู่มือ IC SHIP SUPPORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้